AT89S52MCU 增强实验板　使用说明书
V2.0

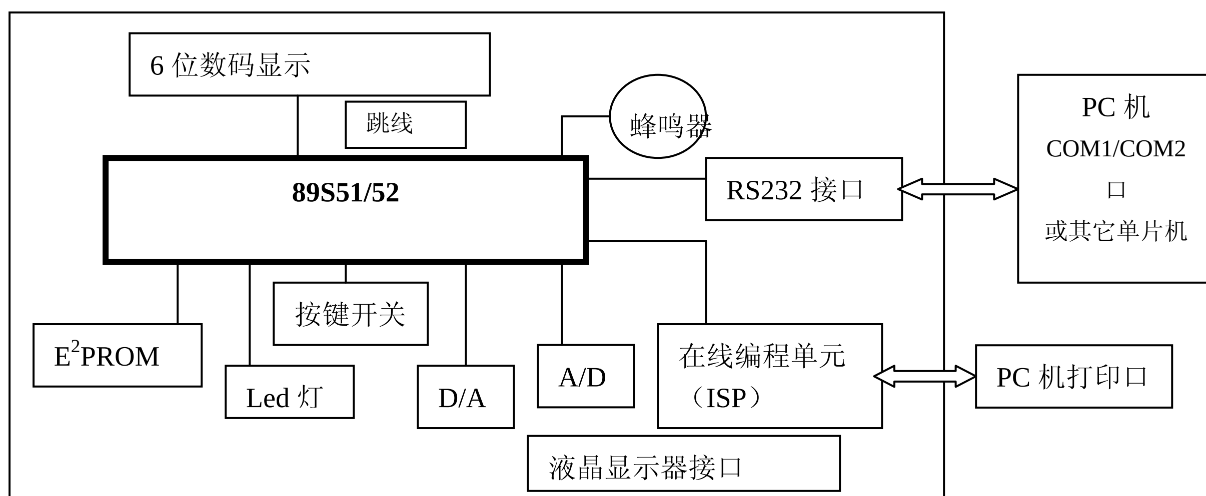
2006-4

目录

第一章 电路结构与元器件分布

1．1　单片机可在线编程多功能实验板的结构框图

单片机可在线编程多功能实验板的结构框图如图一所示。



图一　实验板结构框图

实验板结合了单片机在线编程（烧写）功能及程序运行功能，使得用户一板在手便拥有了编程器和实验板两套设备。通过带锁按压开关方便地实现系统编程状态和程序运行状态之间的转换，马上能观察编程的运行结果，从而免去了单片机开发中必须的昂贵的硬件仿真器和专用编程器的开销。同时由于在线编程，不用频繁反复拔、插单片机，节省了时间，减少了损耗。

同时本实验板，已经把下载电路完整的做在了电路板中，不再需要专用下载线，只需要普通并口线就可以下载单片机程序，这又为消费者节省了不少金钱。

我们保证所设计的电路稳定，在实验过程中不需要插拔任何电线/跳线，单片机所有引脚资源可用，不存在保密的从处理器，下载过程方便快捷。

另外，本实验板采用 usb 供电，用户只需要将普通 usb 延长线插入板子接口即可实验，保证实验过程不需要任何实验室中才能得到设备，学生完全可以在家中/寝室中完成全部实验。

利用单片机的各单个内部资源，能开发出如下基本硬件实验：

1、　并行口的输入、输出实验
2、　7 段数码管的显示与控制、键盘编程、LED 灯编程、蜂鸣器编程
3、　串行通信实验（单片机和单片机、单片机和 PC 机的通信）
4、　串行 EEPROM
5、　串行 A/D 转换

6、 串行 D/A 转换

7、 液晶显示器的设计和控制编程（需用户自己设计连线方案）

综合利用上述资源，用户可以设计诸如多功能数字钟、数字电压表、数字频率计，音乐盒,电子琴，频率合成，测频，测相、控制系统等应用系统，可作为以后课程设计和毕业设计和搞创新发明的硬件平台。

实验板已安装的器材有：

6 位数码管（做数字钟可以同时显示小时，分钟，秒）；

24C04 存储器（存储各种用户数据）；

TLC549 模数转换器（A/D：可将模拟电压转换为单片机识别的数字信号）

TLC5615 数模转换器（D/A：可将数字信号转换为模拟电压）

232 串口电路（可以和电脑或其他单片机进行数据通信）；

4 个 LED；（用户可以对其编程以表示各种状态）

蜂鸣器；（可以做报警或提示）

4 键控制按钮；（用户输入用，也可以扩展为 4*4 矩阵键盘）

扩展接口；（将单片机的特殊功能 IO 引出）

232 通讯线；

USB 供电接口；（用户不需要购买直流电源，直接使用 PC 的 USB 口对板子供电）

AT89S52 芯片（采用缩紧座，方便作为通用编程器）；

标准液晶显示器接口（可以接驳 16xx 系列所有液晶显示器）

配套 CD 程序、资料、实验程序等。

AT89S52MCU 多功能实验板适合学校、单片机爱好者、一般开发研究单位使用。

本产品已在中南地区部分高校试用 2 年，实践证明了他的实用可靠，是单片机爱好者的好帮手。

1．2　单片机可在线编程多功能实验板的元件分布

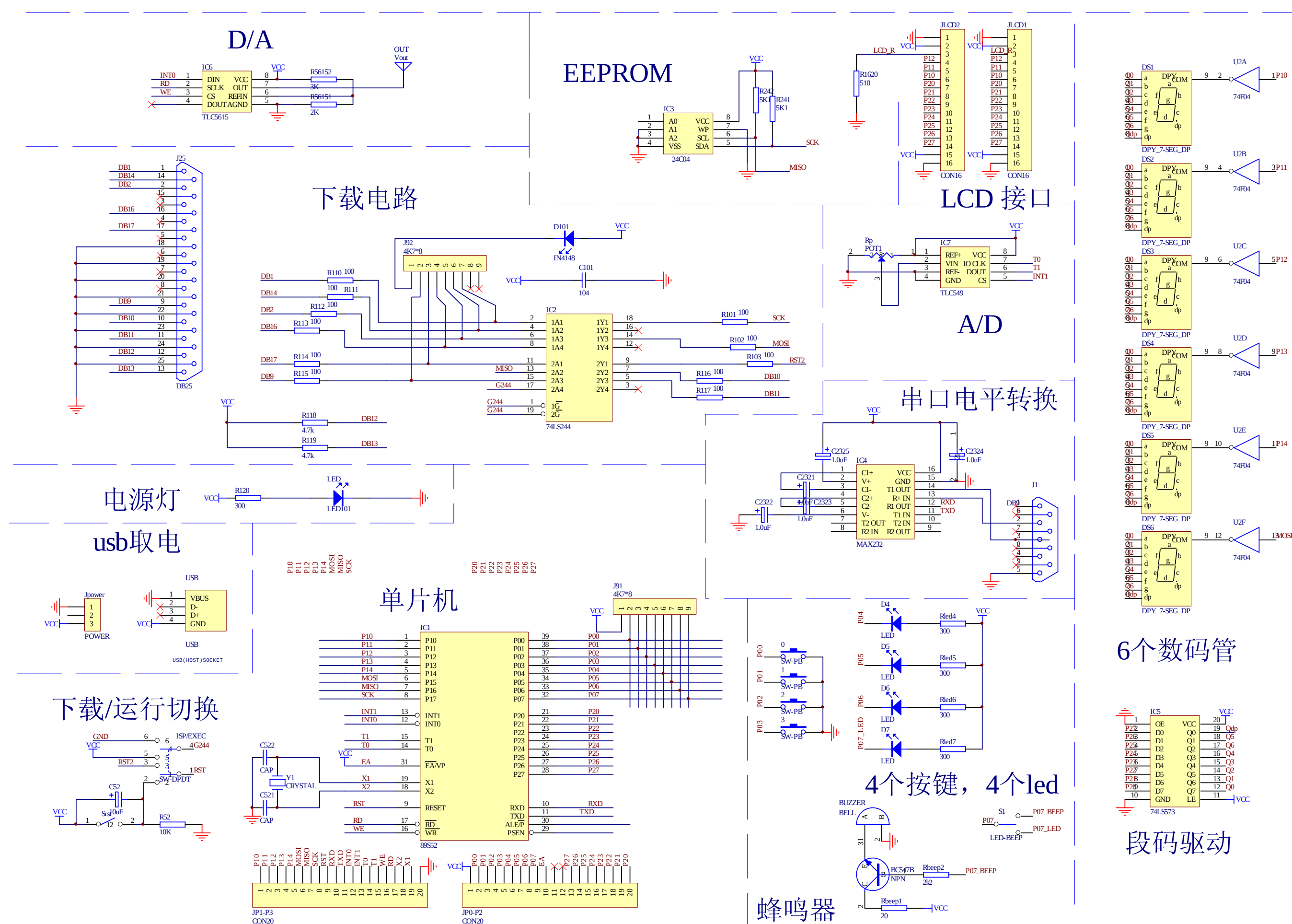
可在线编程单片机多功能实验板的元件分布如图二所示。



图二　实验板元件分布图

1．3 多功能实验板的电路图

单片机可在线编程多功能实验板的电路原理如图三所示.
您可以将文档的显示比例放大到 500%可以看到原理图的细节



图三　实验板原理图

第二章 实验板面各模块功能和测试

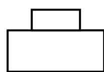
2.1 板上模块说明

1、 带锁按压开关 ISP/EXE

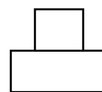
带锁按压开关用于切换系统的工作状态。

当开关**按下**，系统处于编程状态，此时可以通过 Atmel 下载软件下载编译好的程序的十六进制代码（.hex）文件。

当开关**弹起**，系统处于程序运行状态，此时可以运行单片机内部程序。



编程状态，此时可以通过并口下载



运行下载的程序

2、复位按钮 系统中有 1 个复位按钮，用于单片机复位，每按一下，单片机复位一次。

3、四个用户按钮

按钮 0-3 用作输入设备，用于置数或控制，例如用户可以通过此开关，控制单片机的动作。

4、电源指示灯 当系统接通电源，处于工作状态时，二极管亮，否则二极管灭。

5、四个用户 Led 作为输出设备，用户在程序中设定其功能，第 4 个用户 LED 和蜂鸣器公用一个单片机端口，需要用户用跳线选择使用第四个 LED 或蜂鸣器。

6、六个 7 段数码管 作为输出设备，用户在程序中设定其显示值和显示方式，为了使用户得到最好的锻炼，数码管采用动态扫描方式

7、蜂鸣器 作为输出设备，可以作为报警等功能，第 4 个用户 LED 和蜂鸣器公用一个单片机端口，需要用户用跳线选择使用第四个 LED 或蜂鸣器。

8、EEPROM 作为存储数据的设备，系统断电后仍能保存数据

9、A/D 作为输入设备，可以把模拟电压值变为单片机可以识别的数字值，配合各种传感器就可以构成复杂的系统。

10、D/A 作为输出设备，可以把数字值变为电压值，是构成较为复杂系统不可缺少的部分。

11、通信串口 作为输入输出设备，可以和其他的各种单片机或 PC、笔记本电脑、掌上电脑进行通信

12、USB 供电接口 作为系统电源的接入口，系统不需要用稳压电源供电，更加方便实用，插入电脑的 USB 接口，或者 USB 延长线接口就可以使系统供电

13、并口下载电路 系统采用并口线下载单片机程序，使用 Atmel 最稳定的下载方式

14、 液晶显示器接口 采用校准液晶显示器接口，可以接驳 1601,1602,1620 系列所有型号的液晶显示器，我们提供一个 1620 液晶显示器的接口程序，此功能为高级用户选用的功能。

15、跳线 跳线的方向决定了单片机的 P0.7 口控制第四个 led 灯（跳线选择左边）还是控制蜂鸣器 beep（跳线选择右边）

2. 2 测试各个模块的功能

您在购买本板时，一个综合测试程序已经在单片机内

以下介绍如何使用这个测试程序。

综合测试的机器码程序（TASTALL.HEX）用于测试你的实验板是否完好，在芯片齐备的情况下，完成下面的 5 个实验项目的测试，当你购买实验板时 ，该程序已被下载进单片机。

在执行 TASTALL.HEX 时，按压开关 W1 使处于弹起状态。

用户可以按住按钮 0-3 的不同状态复位系统，选择运行单片机中的各个测试子程序

1、按钮 0—3 都不按下，复位系统：

测试用户按纽和用户 led，系统上电后，数码管全亮，按下用户按钮，相应的 led 亮，如果跳线选择蜂鸣器一边，按下第四个按钮，蜂鸣器鸣叫；同时液晶屏显示欢迎画面

2、按下按钮 0，并复位系统：

数码管跑马，测试数码管

3、按下按钮 1，并复位系统：

测试 EEPROM，读写成功数码管有变化的显示

4、按下按钮 2，并复位系统：

测试 A/D 转换器，旋动电位器，他会改变 A/D 芯片的输入电压，A/D 芯片把模拟电压转换为数字量，单片机将数字量显示出来，其中 0v 电压对应数字量 00，5V 电压对应数字量 FF（十六进制）

5、按下按钮 3，并复位系统：

测试 D/A 转换器，此时将一个示波器的探头连接到 D/A 芯片旁边的 Vout 孔，您可以看到示波器上显示一个频率约为 200Hz 的正弦波形。这是单片机将波形表中的数字量，按一定的周期送到 D/A 转换器进行转换得到的模拟电压变化的波形。

第三章 下载程序详细说明

完成单片机程序下载，需要的软件是：
在线编程软件 Microcontroller ISP Software
我们提供综合测试程序的 16 进制代码（光盘中的 TESTALL.HEX），可以直接下载

3.1 运用在线编程软件 Microcontroller ISP Software 实现在线编程

将汇编或 C 语言编好了源程序后，通过编译器编译、连接、生成了一个 16 进制文件后缀名为.hex（这个过程在下一个章节介绍）。应用多功能实验板和 Microcontroller ISP Software 软件将此.hex 文件烧写进板上的单片机内，即进行在线编程，下载程序步骤如下：

1. 在电脑关机的状态下，将 25 针并口连接线和电脑并口连接好。
   在实验板未上电的情况下，将 25 针并口线的另一头和此实验板的 25 针接口相连，接着接通系统电源，打开电脑。
2. 按下带锁按压开关，使系统处于编程状态，打开 Microcontroller ISP Software 软件，即可进行在线编程。
3. 在程序下载成功之后,再次按带锁按压开关，使之弹起，单片机就可以处于运行状态。

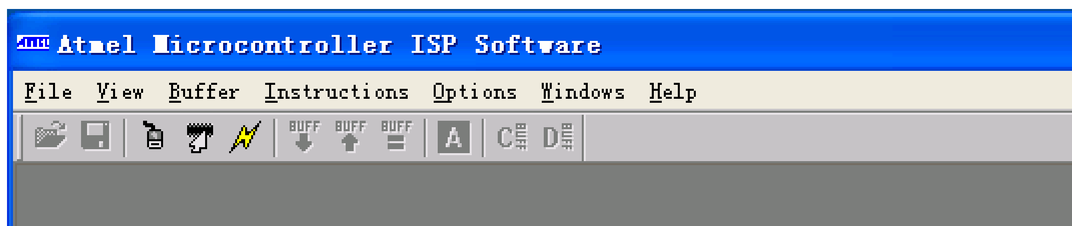
在下载程序之前，大家一定要注意如下事项：

1、连接电脑的 25 针连接线必须在电脑关机状态（断电状态）下安装或者拆除，在电脑 2、运行的情况下，可能给电脑造成伤害。25 针连接线连接单片机系统这一端的接口必须在单片机系统断电的情况下连接或者拆除，否则会给系统带来伤害。

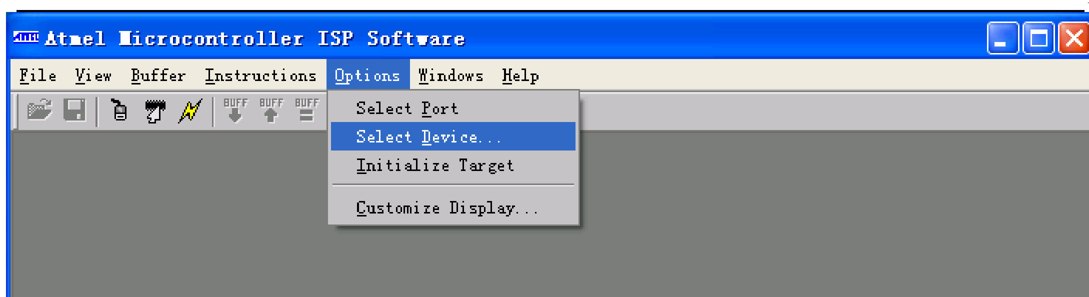
ISP 软件操作步骤：

Microcontroller ISP Software（软件在购买实验板时一并附送）也可在网上下载，网址为
http://www.atmel.com/dyn/resources/prod_documents/at89isp.zip
进行 SETUP 安装后便可使用

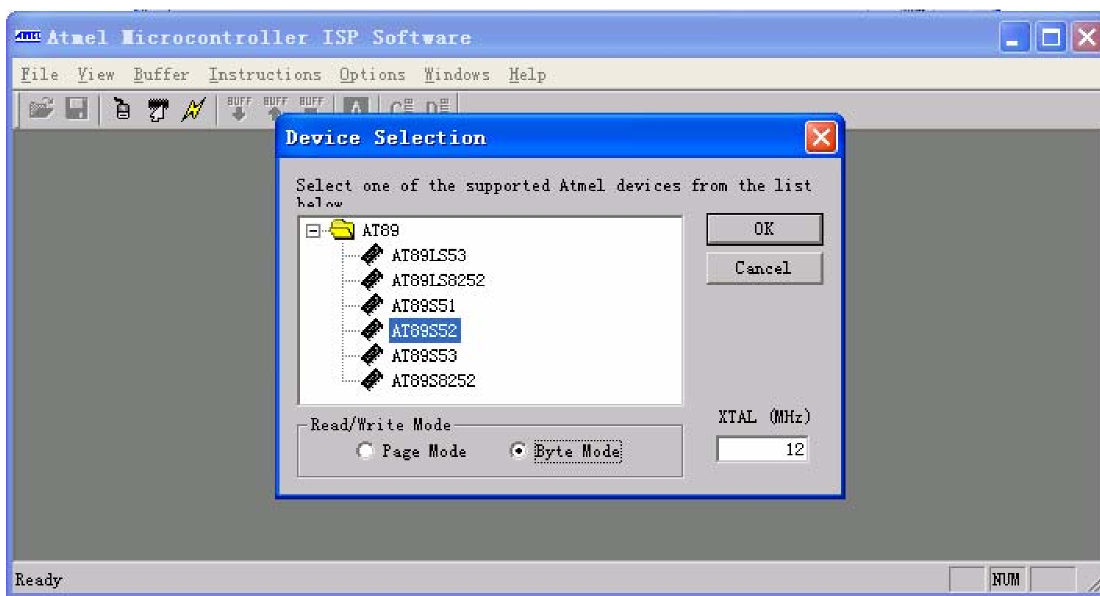
1、 打开 Microcontroller ISP Software，如下图：



2、点击菜单中的 options 选项中的 select device（选择器件）：



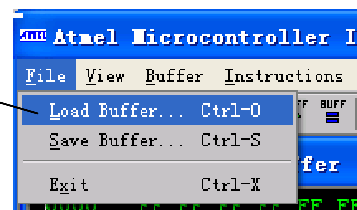
弹出以下窗口：



选择 AT89S52，并选 **Byte Mode** （字节模式）点击“OK”。

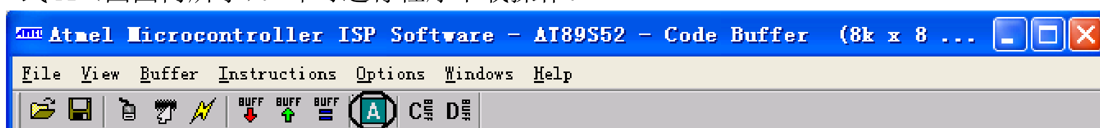
3、点击 options 选项中的 Initialize Target（初始化器件）或者闪电图像的快捷键（图中圆圈所示）。闪电图像的快捷键





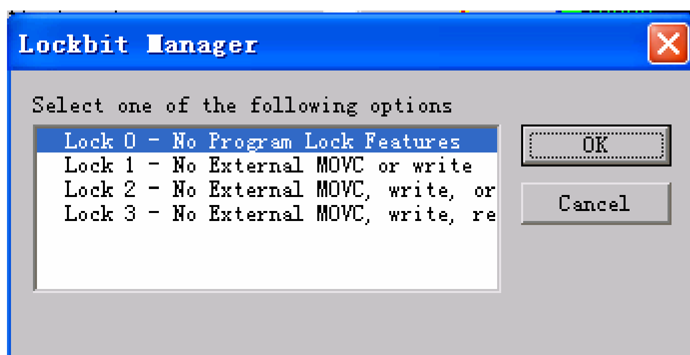
4、接着点击菜单 File 中的 Load Buffer，选择需要下载的程序(经过编译生成的.HEX 十六进制文件)

5、程序选定之后即可进行下载操作，点击 Instructions 中的 Auto Program 或者点击快捷方式 A（圆圈内所示）。即可进行程序下载操作。



经过以上步骤，Microcontroller ISP Software 软件便对系统进行在线编程。在编程成功之后，

出现如下对话框：



可选择“OK”或者“Cancel”以决定是否对单片机上锁（加密）。至此，程序下载结束。

6、进入程序运行状态　运行下载到单片机中的程序。有两种方式

方式一 点击菜单 Instruction 的 Run Targe。用 ISP 软件启动单片机程序,此法运行单片机中的程序时由于单片机还和编程器连接，如果你要使用 P1.5、P1.6、P1.7 口线，它们将工作不正常，建议使用下面的方式二。如果你不使用 P1.5、P1.6、P1.7 口线，可在此状态下运行程序。

方式二 按一下按压锁紧开关，使之处于弹起状态，系统即与板的编程器部分断开，进入程序运行状态，按复位按钮后，运行单片机中的程序。

注意：

1，每次在对单片机进行编程操作之前都得初始化系统：点击 options 选项中的 Initialize Target（初始化器件）或者闪电图像快捷键。

2，只要您总是按照“初始化--〉自动下载”（即上面的 3，4，5 步骤）操作，芯片肯定可以很稳定的下载，请不要使用下载软件中的其他功能，以造成单片机进入不正常的状态

3，有些兼容机上可能无法下载，可以在系统 CMOS 设置中将 onborad parallel port 一项（不同的 BIOS 此项的位置不同，请用户自行找到）设置为 EPP 方式（原始状态可能为 SPP），对于 IBM 的笔记本电脑请查看并口占用的中断和端口号和普通 PC 相同，可以解决问题。我们对 Win98，Win2000，WinXP 都做过测试，均能顺利下载程序。

第四章　单片机仿真调试集成软件包 Wave 的使用

Wave 是南京伟福公司的仿真调试集成软件包，它集编辑、编译、连接、跟踪调试于一体，是单片机开发的必须软件。程序在烧写进单片机前，必须将它送入计算机（编辑)、编译（或汇编)、连接，生成.HEX 文件，经过模拟运行（单步执行、断点执行等)，即软件仿真调试，无误后再烧写，这样可提高开发效率。该软件包即可对汇编源程序进行调试，也可对 C 语言程序进行调试。

4.1　软件的安装和设置

Wave软件包（随实验板附送）可在网站下载，网址为http://www.wave-cn.com/，下载的文件是E6000(或E2000),它自带汇编器.，

经过解压后，执行 SETUP.EXE,按照安装程序的提示,输入相应的内容,直至安装结束。

在安装过程中，如果用户没有指定安装目录，系统会在 C 盘建立一个 C:\WAVE6000 目录(文件夹),结构如下:

目录	内容
C:\WAVE6000	
BIN	可执行程序及相关配置文件
HELP	帮助文件和使用说明
SAMPLES	样例和演示程序

汇编源程序的汇编器

伟福编译系统已内嵌汇编编译器(伟福编译器),用户可直接应用 WAVE 对汇编源程序进行汇编，

- C 语言程序的编译器

WAVE 自身不带C51 编译器，可以从网上下载Keil，也可以从WAVE 网站下载ICExplorerD，解压后将其中的 P51 文件拷到 COMP51 目录下，并运行 P51 自解压。（光盘里也带这个包 Comp51.RAR）

安装 C51 系列 CPU 的编译器

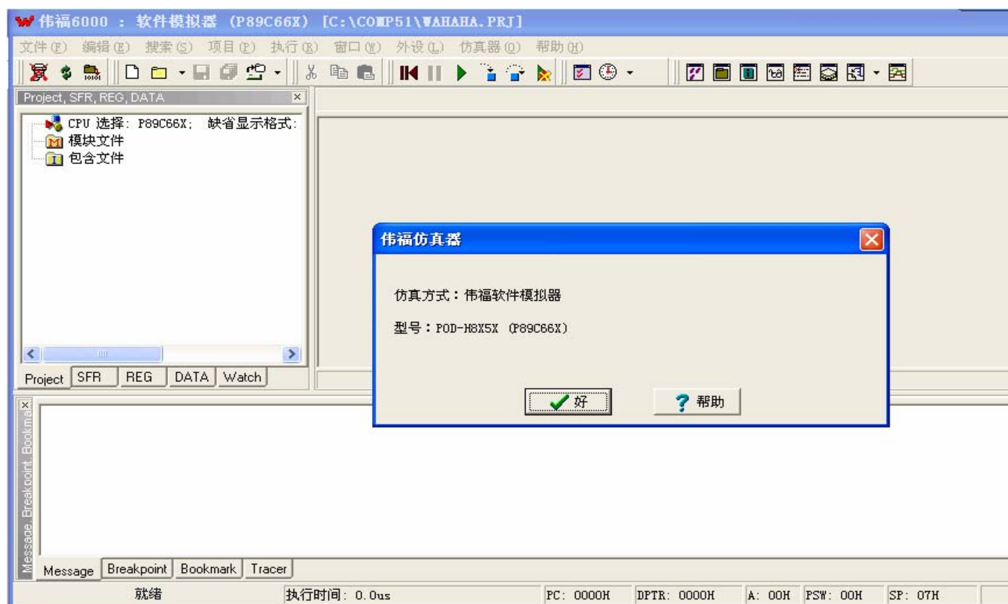
1. 进入 C:\盘根目录,建立 C:\COMP51 子目录(文件夹).
2. 将 Comp51.rar 解压后把 C51 目录复制到 C:\COMP51 子目录(文件夹).
3. 在[主菜单] / 仿真器 / 仿真器设置 / 语言 对话框的[编译器路径]指定为 C:\COMP51\C51

如果用户将 C51 编译器安装在硬盘其他地方,请在[编译器路径]指明其位置.

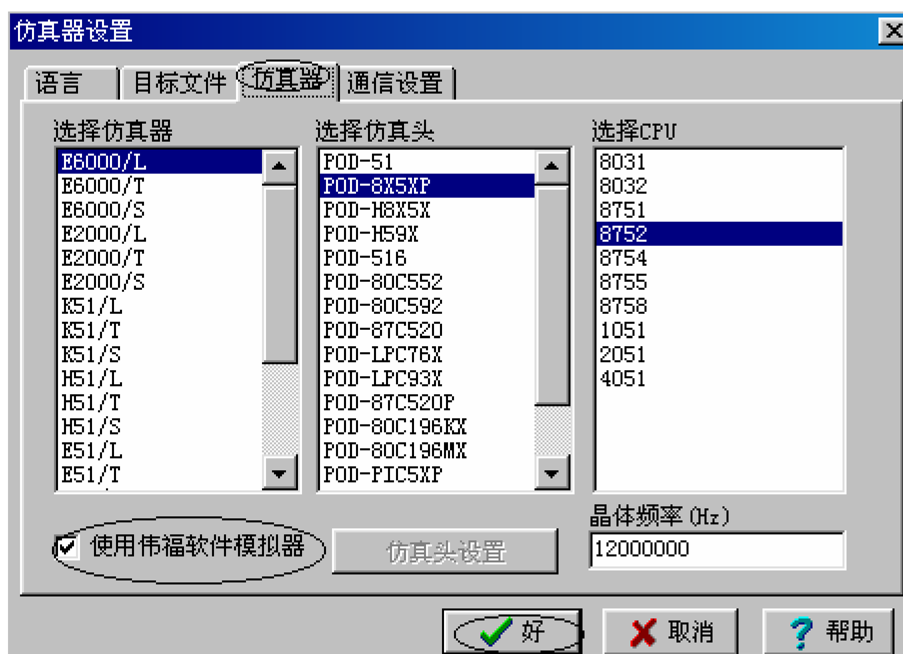
- 注意：如果未安装第三方C51 编译器，或未将主菜单中的【 仿真器 / 仿真器设置 / 语言[编译器选择单选框]中的编译器设置正确，则编译时会出现错误。

经过以上步骤,我们就可以使用 Wave 对程序进行编译，生成我们所需要的.HEX 文件.下面对 Wave 的基本操作做一些简介.

1. 打开 Wave 编译器,出现如下界面,并在其中点击“好”。



2．点击菜单栏文件选项中的“仿真器”出现如下界面，并点击其中 “使用伟福软件模拟器”，使其中出现“√”，“选择 CPU ”选项中的”8751 或 8752（取决你是用 89S51 还是用 89S52，它们和 8031、8032 是兼容的）。由于不使用仿真器作硬件仿真，因此“选择仿真器”“选择仿真头”以及“通信设置” “仿真头设置”无需选择。然后点击“好”



在“仿真器设置”/语言 中选择伟福汇编器，如用 C 语言置好 C 编译器路径见下图：

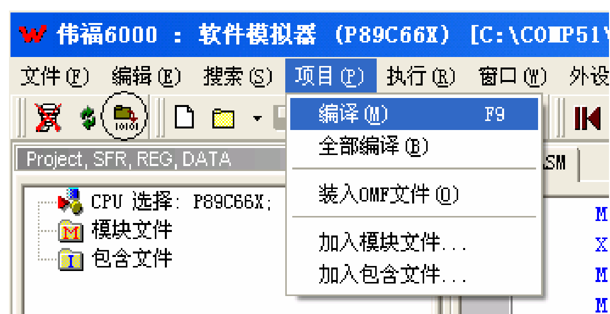


4.2 编辑和编译程序

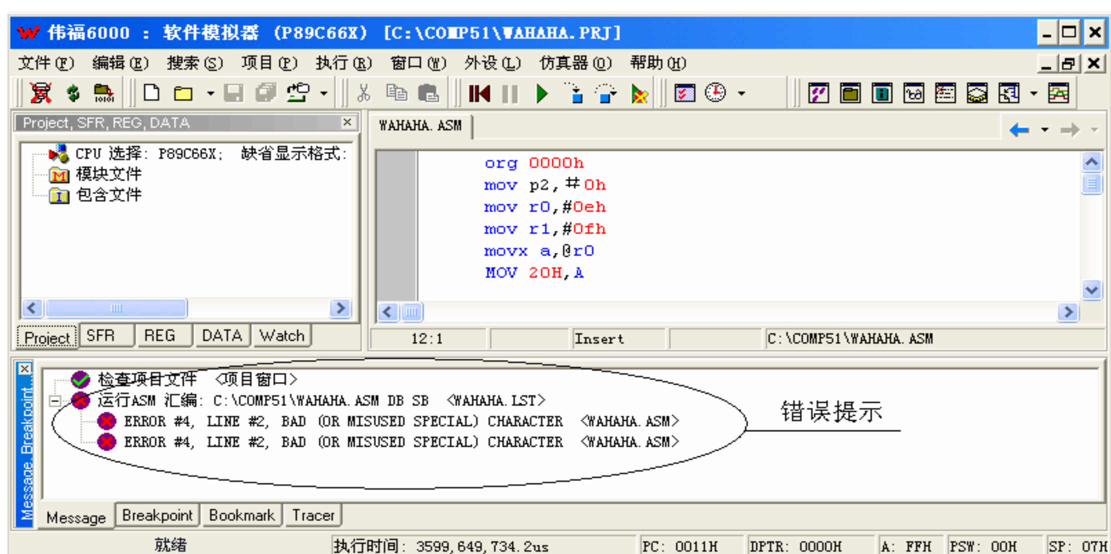
编辑程序或者打开已编好的程序,点击菜单栏文件选项中的“新建文件”编写程序或者点击“打开文件”打开已经编好的程序。**注意**:如果是汇编语言程序以·ASM 后缀存盘,如果是 C 语言程序以·C 后缀存盘。



2、对源文件进行编译。点击菜单“项目”中的“编译”,或者点击如图所示的快捷方式,就可以对程序进行编译,并生成.HEX 文件。(汇编和 C 语言均如此)

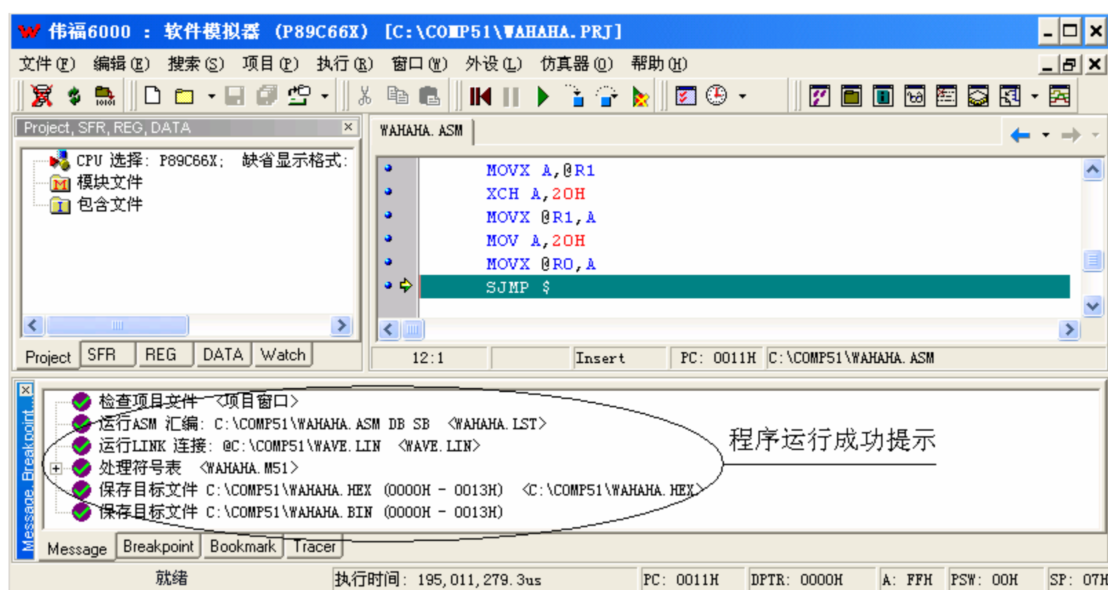


如果程序存在语法或者结构错误,在程序编译时会出现错误提示。如下图所示,大家可以根据错误行提示修改程序错误之处。



如果程序能够成功运行，对话框中就会出现运行成功提示，并生成 ·HEX 文件如下所

示：



<u>注意：</u>当一个程序由多个模块组合时（例如C语言和汇编的混和编程或同一语言的多个模块组合），必须建立一个工程项目，方法是：

1) 点“文件-〉新建项目”
2) 在“加入模块文件对话框”将已经保存好的原文件加入，
3）含文件对话框”中加入包含文件，(如果没有包含文件可以点取消)。
4) 在“保存项目”对话框中键入项目的名字
5) 点击项目菜单中的编译

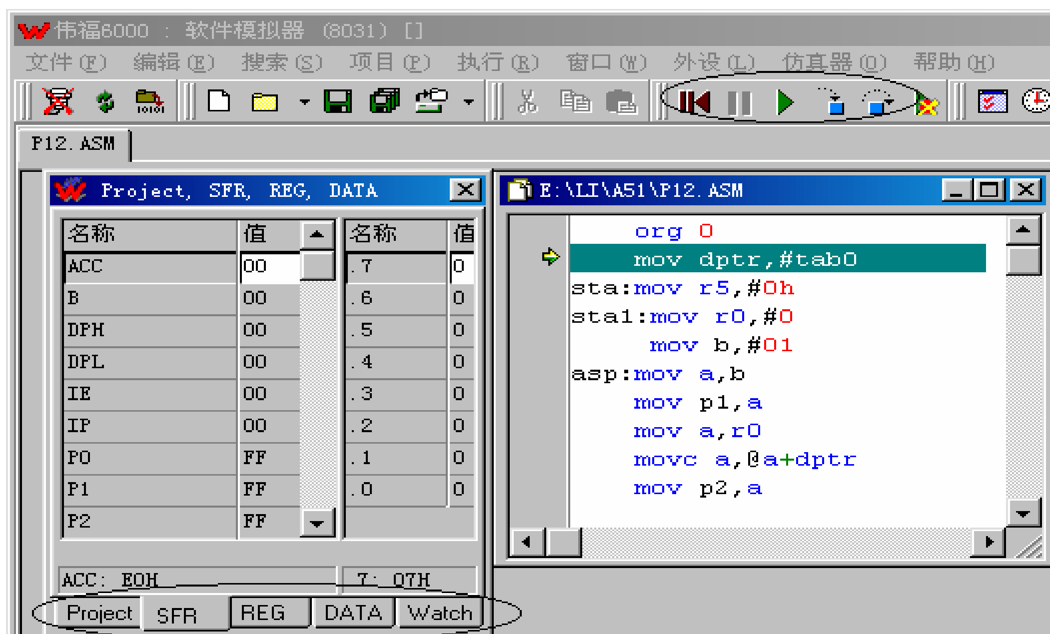
点击菜单“执行”中的各个选项，或者图中所示快捷方式运行程序。

经过以上操作，我们就可以得到单片机在线编程所需要的.HEX 文件，该文件即为系统在线编程的烧写文件。

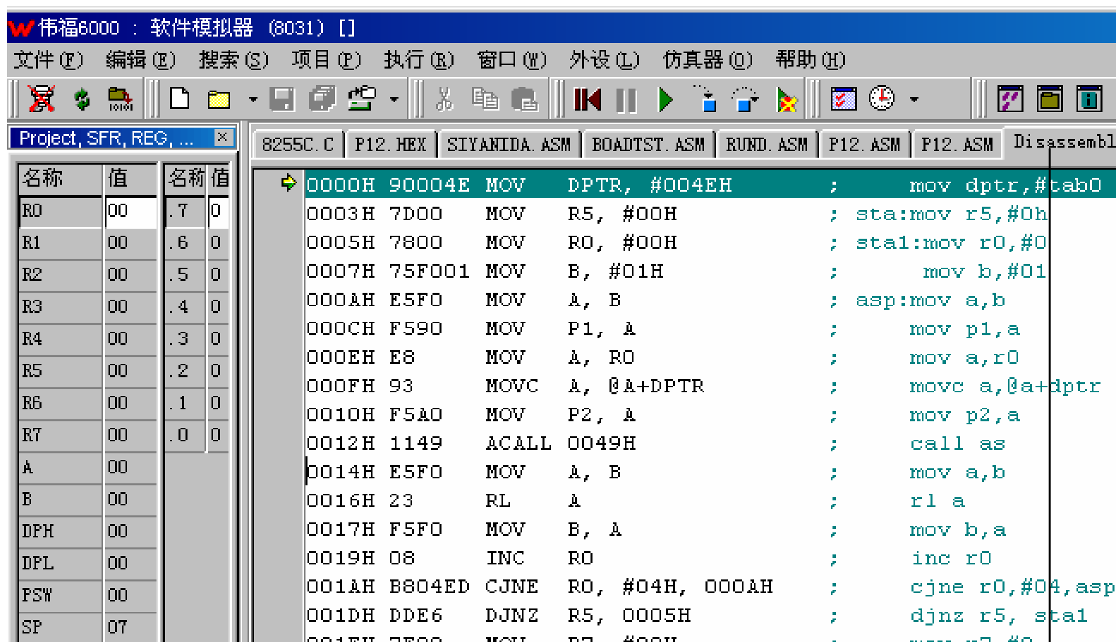
4.3　仿真调试程序

编译好的程序可利用 WAVE 软件包模拟执行（仿真调试），用单步跟踪、设置断点、察看单片机的内存和特殊寄存器的内容变化是否合乎设计意图，若有错误，修改程序，重复上述步骤，直至正确，这样可以提高效率。

执行操作方式可在主菜单“执行”挑选，也可点击快捷键（见下图）。该栏中有复位、暂停、连续执行、全速执行（即连续执行）、跟踪执行、单步执行等。在主菜单“窗口”中可以打开 CPU 窗口或数据窗口以观察运行状态。点击观察窗口的不同选择可以分别观察 SFR、REG(R0—R7、…)、DATA(内部数据存储器)、Watch(各标号和各变量的地址)



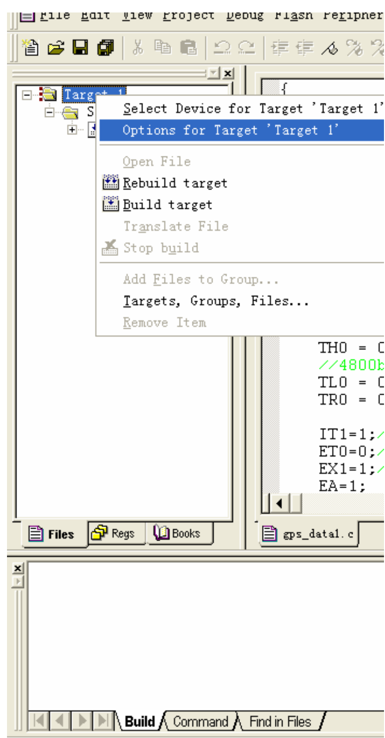
以下是 CPU 窗口



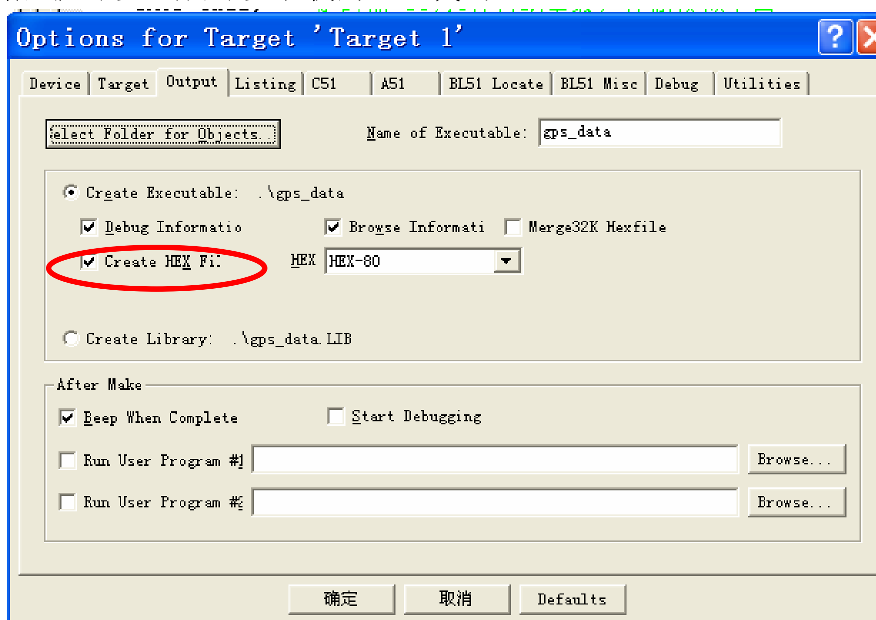
第五章 Keil 编译程序的方法

Keil 是单片机最好的编程环境，由于是商业软件，所以我们只附送一个破解后的测试版本，用于向用户演示编译过程，请用户安装后于 24 小时之内删除，并不能用于任何商业目的。

安装 Keil 后，加载程序项目，右键点击 target,选择 Options for Target



在弹出对话框的 Output 页签中选上 Creat HEX File 选项，这样程序编译成功后就可以生成可以下载的.hex 代码。



Keil 环境的详细使用方法，请参看附送的“Keil 中文教程.pdf”文档

第六章　基础实验

实验一　单片机程序编译调试基础[实验编号 001]

一、实验目的

① 学习 windows 平台下 wave 仿真软件的编辑、编译、排错、调试方式

② 学习单步、断点、自动跟踪/单步、全速执行等各种执行方法。

注意：这个程序没有访问任何单片机的硬件端口，也不需要下载调试。

二、实验步骤

1. 运行 WAVE 软件，点击菜单栏选项中的“仿真器”，并点击其中 “使用伟福软件模拟器”，使其中出现“√”，“选择 CPU ”选项中的 8031 或 8032（取决你是用 89S51 还是用 89S52），点击 “√好”，即进入软件模拟环境。

2. 鼠标点击菜单“文件（F）” →“新建文件（N）”，进入编辑窗口键入以下程序：（此程序在光盘中为 001.asm，光盘中还有一个与之功能相同的 001.c 文件。）

以下是汇编语言的程序[文件名：001.asm]

```
        INDEX    EQU   20H
        SUM      EQU   21H

        ORG     0000H
START： MOV   INDEX，  #5
        MOV   A，#0
LOOP：  ADD   A，INDEX
        DJNZ  INDEX，LOOP
        MOV   SUM，A
        SJMP  START
        END
```

以 **001.asm** 文件存盘。注意，请为每个程序建立一个文件夹，这样便于管理，而且文件夹路径中不要有中文名称。

3. 点击快捷菜单编译（或按 F9 键），伟福仿真器自带的汇编器对源程序汇编，会在这个文件夹下生成 **001.hex** 文件。

4. 点击快捷菜单，打开 CPU 窗口，打开内部数据存储器 DATA 窗口，调整窗口大小，和窗口位置，使 DATA 窗口、CPU 窗口能同时显示在屏幕上。

5. 自动跟踪/单步 执行程序

鼠标点击菜单“执行（R）” →“自动跟踪/单步” ，可以看到程序自动单步执行，同时可以观察 A 累加器内容的变化及内部数据存储器 DATA 窗口中内容的变化。

6. 利用各种操作方式（单步、全速）逐条执行观察 A、21H 单元、20H 单元内容的变化。

注意：单片机程序既可以用汇编语言（asm）也可以用 C 语言编写，用户可以分别学习。在编写 C 语言文件时，需要将文件后缀名保存为.C 文件。

以下是 C 语言的程序[文件名：001.c]

```
main()
{
    char *index=0x20;              /*index  指向内部 RAM 20H 单元*/
    char *sum=0x21;        /*sum 指向内部 RAM 21H 单元*/
    *index=5;
    *sum=0;
    for(; *index ; (*index)--)
    {
        *sum = *sum + *index;        /*sum 存放累加和*/
      }
}
```

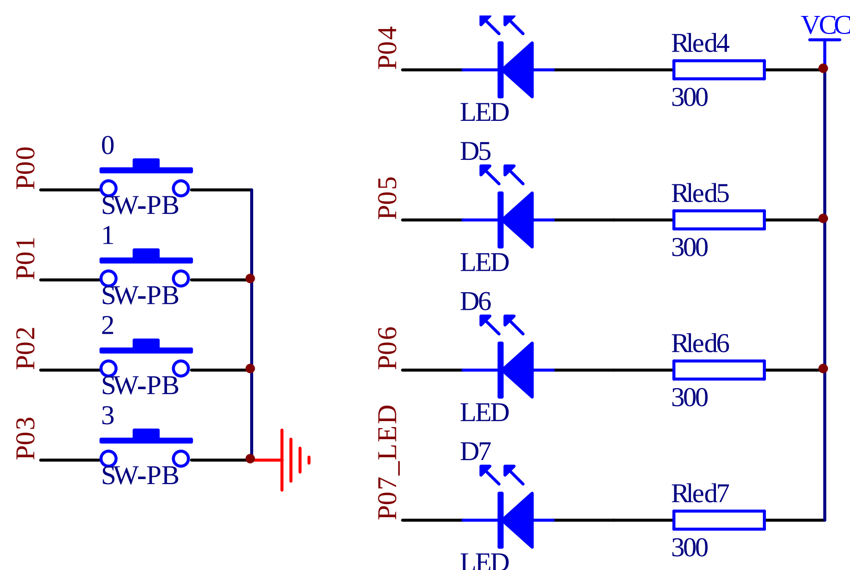
实验二　学会控制单片机的硬件端口[实验编号 002]

一、实验目的

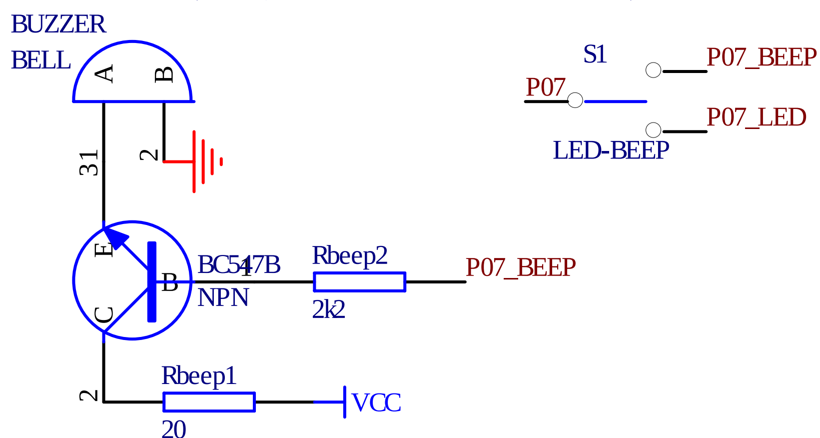
1、熟悉 51 单片机端口的输入方式，输出方式的编程
2、熟悉 51 单片机端口应用编程软件仿真调试方法
3、学会下载单片机程序（在线编程 ISP）方法

二、实验电路和程序

实验电路



4个按键，4个led



蜂鸣器

实验程序

单片机的 P0.0，P0.1，P0.2，P0.3 对应了 4 个按键开关，P0.4，P0.5，P0.6，P0.7 对应了 4 个发光二极管 LED，我们这个程序的任务是：按开关，控制对应的灯亮；如果 4 个开关同时按下，数码管的小数点（连到单片机的 P2.7）熄灭。

以下是汇编语言的程序[文件名：002.asm]

```
      ORG     0000H
      MOV     P0,#0FFH
      MOV     B,P0
      MOV     A,P0
L1:   CJNE    A,B,NEXT    ;等待按键
      SJMP    L1
NEXT:   CJNE    A,#0F0H,LED;4 个键是否都按下？
      ANL     P2,#7FH     ;是，则 p2.7 复位
LED:    MOV     A,P0
      SWAP    A         ;高低 4 位互换
      MOV     P0,A
      SJMP        L1
      END
```

以下是 C 语言的程序[文件名：002.c]

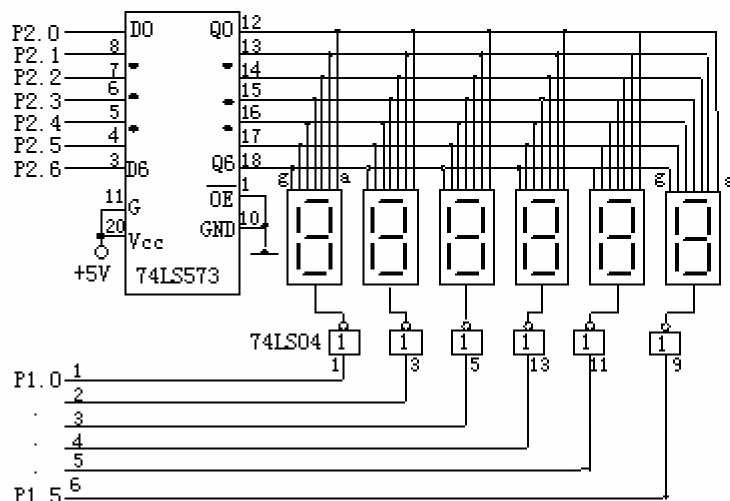
```
#include<reg51.h>
void main()
{
      P0=0xff;                //在读按键状态前需要先对端口写高电平
      P0=P0<<4 ;              //将按键的状态移位对应到 LED 灯的状态
      if((P0|0x0f)==0x0f)     //
          P2&=0x7f;
}
```

实验三　用单片机的控制数码管显示数字[实验编号 003]

一，实验目的

进一步掌握控制单片机端口的方法，并掌握 7 段数码管的使用方法。

二，实验电路图



三，实验程序

数码跑马程序：六个数码管从左至右依次显示 1，然后依次显示 2，……依次显示 F。0～F 都显示完后，再从 0 开始显示，如此循环。

以下是汇编语言的程序[文件名：003.asm]

```
DLED:    MOV   R0,#0                  ;R0 存字形表偏移量
  WE:    MOV   A,#01                  ;A 置数码管位选代码
NEXT:    MOV   B,A                    ;保存位选代码
         MOV   P1,A
         MOV   DPTR,#TAB0             ;DPTR 置字形表头地址;
      MOV   A, R0
         MOVC    A,@A+DPTR            ;查字形码表
         MOV   P2,A                   ;送 P2 口输出
      MOV   R3,#0                     ;延时
 LOP:    MOV   R4,#0
LOP1:    NOP
         NOP
         DJNZ  R4,LOP1
         DJNZ  R3,LOP                 ;延时
         MOV   A,B
         RL    A                      ;指向下一个数码管
         CJNE  A,#40H, NEXT           ;六个数码管显示完否
         INC   R0                     ;指向下一位字形
         CJNE  R0,#10H,WE             ;从 0 到 F 显示完否
         SJMP  DLED
TAB0:   DB
     0BFH,96H,0DBH,0CFH,0E6H,0EDH,0FDH,087H,0FFH,0EFH,0F7H,0FCH,0B9H,0DEH,0F9H,
0F1H
      END
```

以下是 C 语言的程序**[文件名：003.c]**

```
#include <reg51.h>
#define uchar unsigned char
#define uint unsigned int
main()
{
        uchar i, k;
        uint j;
        uchar code
tab[16]={0xBF,0x96,0xDB,0xCF,0xE6,0xED,0xFD,0x87,0xFF,0xEF,0xF7,0xFC,0xB9,0xDE,0xF9,0x
71};
    while(1)
      {
                  for(i=0;i<16;i++)                    /*依次显示 0～F*/
        {
              P1=0x01;                                     /*位选*/
            P2=tab[i];
              for(k=0;k<7;k++)                             /*六个数码管轮流显示*/
                  {

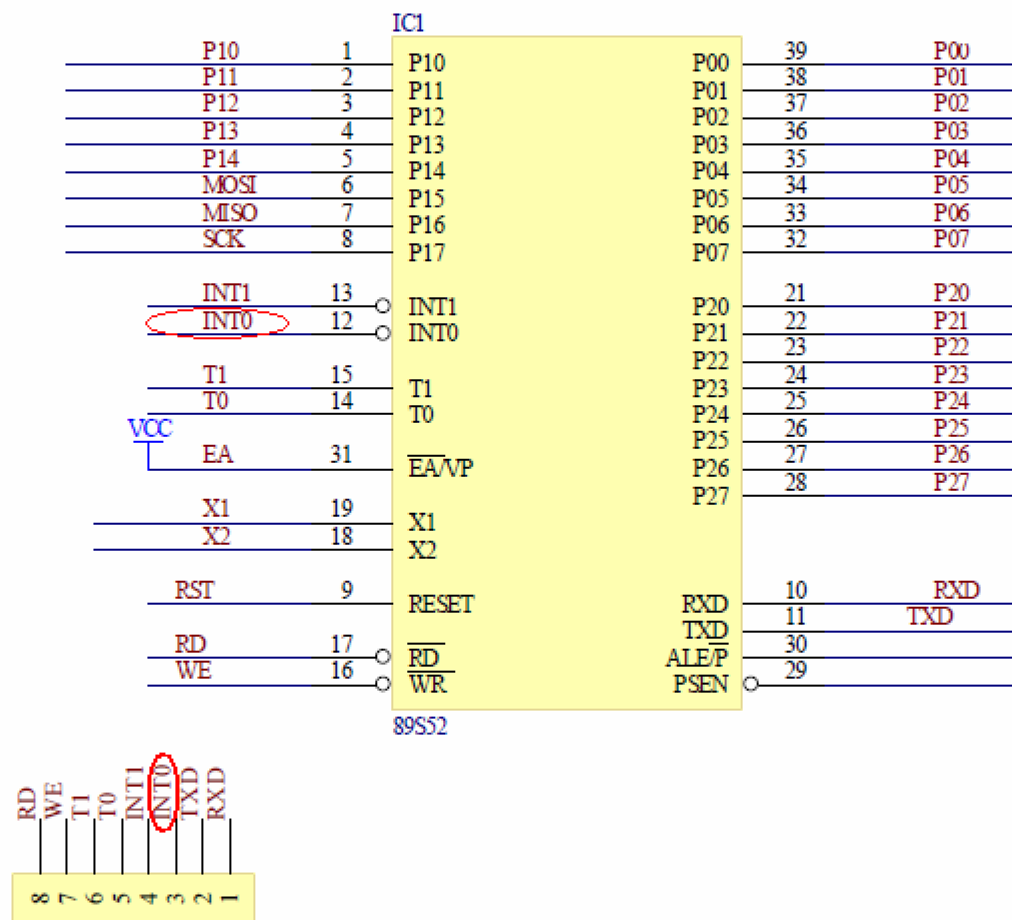
                              P1<<=1;
                              for( j=0; j<25000; j++ );        /*延时*/
                        }
            }
        }
}
```

实验四　单片机的中断程序[实验编号 004]

实验目的

了解中断的产生及响应过程，掌握中断程序的编制。

二、　实验连线



用户需要将中断信号产生源接入单片机的 INT0 引脚（P3.2)。建议大家用信号源的 TTL 电平输入，或者用另一个单片机产生跳变的信号接入单片机的 INT0 引脚。

三、实验程序

以下是汇编语言的程序[文件名：**004.asm]**

```
         ORG 0000H
         AJMP STAR
         ORG 0003H          ；中断服务
         RL A
         MOV    P2,A
         RETI
   STAR: MOV P1,#04H   ；第三个数码管亮
         MOV A,#01H
         MOV P2,A
         SETB EA      ；置 EA=1
         SETB EX0     ；允许 INT0 中断,
         SETB IT0     ；边缘触发中断
         SJMP $
```

①分析该程序的功能及实验现象

②将该程序烧进单片机、运行，观察执行的现象是否和估计一致

以下是 C 语言的程序[文件名：004.c]

```
#include <reg51.h>
#define uchar unsigned char

uchar a;

int0() interrupt 0 using 0
{
    a<<=1;              /*a 左移一位，使数码管的下一段亮*/
    if (a==0)
      {
        a=0x01;
      }
    P2=a|0x80;
}

main()
{
        P1=0x04;        /*第三个数码管亮*/
        a=0x01;
        P2=a|0x80;          /*使数码管的第一段亮*/
        EA=1;           /*置 EA=1*/
        EX0=1;          /*允许 INT0 中断*/
        IT0=1;      /*边缘触发中断*/
        while(1);       /*等中断*/
}
```

实验五　串行通信实验[实验编号 005]

一、实验目的

掌握单片机串行通信的工作原理及编程方法。

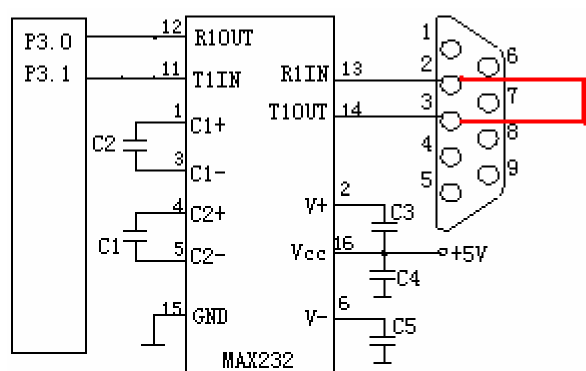
二、实验连线和程序

单片机的 P3.0（RXD），,P3.1（TXD）通过电平转换芯片 MAX232 连到 9 针 D 型插座上，通过 9 针 D 型插座和电缆可以与单片机、PC 机进行串行通信实验。

注意：在插拔串口前设备必须关电！PC 插拔串口线时必须关机！

三、单机自发自收通信实验

实验电路如下图，用导线将 D 型插头的 2、3 引脚连起来（见下图红线）



单片机自发自收通信实验实验连线图

实验程序

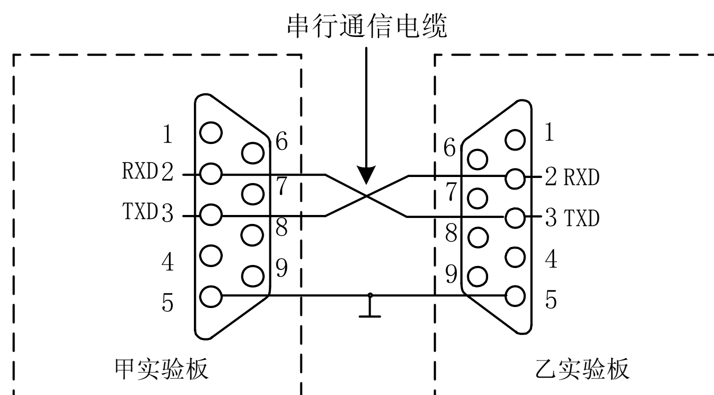
```
          ORG     0000H
     TIRI :MOV    TMOD,   #20H
          MOV     TH1,    #0CCH
          MOV     TL1,    #0CCH          ; 设定波特率
          SETB    TR1
          MOV     R0,     #0
          MOV     SCON,   #50H           ; 方式 1 发送，允许接收
ABC:      CLR     TI
          MOV     P1 ,    #0FFH
          LCALL       DAY1
          MOV     A ,     R0
          MOV     SBUF ,  A               ;发送
          INC     R0
          CJNE    R0,     #10H,    RGIS
          MOV     R0,     #0
RGIS:     JNB     RI,     $
          CLR     RI
          MOV     A,      SBUF           ;接收
          MOV     DPTR,   #LEDAB
          MOVC    A,      @A+DPTR
          MOV     P2,     A              ;显示
          ACALL DAY1
          JNB     TI,     $
          LJMP    ABC
DAY1:     MOV     R4,     #04
DA1:      MOV     R3,     #0
NB :      MOV     R1,     #0
NA:       DJNZ    R1,     NA
          DJNZ    R3,     NB
          DJNZ    R4,     DA1
          RET
LEDAB: DB 06H,5BH,4FH,66H,6DH,7DH,07H,7fh,6fh,77h,7ch,39h,5eh,79h,71h
        END
```

①分析该程序的功能及实验现象

②上机实验，观察的现象是否和估计一致

四、双机通信实验

本实验需要甲、乙两个实验平台，用串行通信电缆线将两个实验平台的 9 针 D 型座连接起来（连线见下图）：



双机通信实验连线图

实验：设甲、乙两机进行通信，波特率为2400，晶振均采用12MHz，甲机将内部数据存储器20H～2FH共16个单元的内容向乙机发送，发送数据之前将数据块长度发给乙机，当数据发送完向乙机发送一个累加效验和。乙机接收数据进行累加和效验，如果和发送方的累加和一致，发数据 0，以示接收正确，否则，发数据FFH，甲方再重发。

程序：

甲机发送程序：**以下是汇编语言的程序[文件名：005A.asm]**

```
TRT:    MOV     TMOD,   #20H
        MOV     TH1,    #0F3H
        MOV     TL1,    #0F3H
        SETB    TR1                 ;T1初始化，使波特率为2400
        MOV     SCON,   #50H        ;串口初始化为方式1，允许接收
RPT:    MOV     R0,     #20H
        MOV     R6,     #10H        ;长度寄存器初始化
        MOV     R5,     #00H        ;效验和寄存器初始化
        MOV     SBUF,   R6      ;发送长度
L1:     JBC     T1,     L2      ;等待发送完
        AJMP    L1
L2:     MOV     A,      @R0         ;读取数据
        MOV     SBUF,   A       ;发送数据
        ADD     A,      R5      ;形成累加和送R5
        MOV     R5,     A
        INC     R0
L4:     JBC     TI,     L3
        AJMP    L4
L3:     DJNZ    R6,     L2      ;判断16个数据是否发送完
        MOV     SBUF,   R5      ;发送效验码
        MOV     R5,     #00H
L6:     JBC     TI,     L5
        AJMP    L6
L5:     JBC     RI,     L7
        AJMP    L5              ;等乙机回答
```

```
L7:  MOV    A,SBUF
     JZ     L8              ;发送正确返回
     AJMP   RPT             ;发送有错，重发
L8:  RET
```

乙机接收程序：**以下是汇编语言的程序[文件名：005B.asm]**

```
RSU:    MOV    TMOD, #20H
        MOV    TH1,  #0F3H
        MOV    TL1,  #0F3H
        SETB   TR1
        MOV    SCON, #50H           ;串行通信方式 1，允许接收
RPT:    MOV    R1,   #20H           ;置接收缓冲区首址
L0:     JBC    RI,   L1
        AJMP   L0
L1:     MOV    A,    SBUF           ;接收数据长度
        MOV    R6,   A
        MOV    R5,   #00H           ;累加和寄存器清 0
WTD:    JBC    RI,   L2
        AJMP   WTD
L2:     MOV    A,    SBUF           ;接收数据
        MOV    @R0,  A
        INC    R0
        ADD    A,    R5
        MOV    R5,   A              ;计算累加效验和
        DJNZ   R6,   WTD            ;未接收完，继续
L5:     JBC    RI,   L4             ;接收对方发来的效验和
        AJMP   L5
L4:     MOV    A,    SBUF
        XRL    A,    R5             ;接收的效验码和计算的效验码是
                                    ;否相同
        MOV    R5,   #00H
        JZ     L6                   ;同，转 L6
        MOV    SBUF, #0FFH          ;不同，出错发送 0FFH
L8:     JBC    TI,   L7
        AJMP   L8                   ;等待 0FFH 发送完
L7:     AJMP   RPT                  ;重新接收
L6:     MOV    SBUF, #00H           ;正确，发送 00H
L9:     JBC    TI,   L10
        AJMP   L9                   ;等待 00H 发送完
L10:    RET
```

甲机发送程序：**以下是 C 语言的程序[文件名：005A.c]**

```
#include <reg51.h>
#define uchar unsigned char
uchar *p=0x20;             /*指针 p 指向内部 RAM 20h 单元*/
void com_init()            /*串口初始化程序*/
{
    TMOD=0x20;
    TH1=0xfa;
    TL1=0xfa;
    TR1=1;
    SCON=0x50;
}

send()
{
   uchar i;                /*i 为发送数据的长度*/
   uchar j=0;          /*j 为效验和*/
sendDT:  i=16;
      SBUF=i;              /*发送长度*/
      while(TI !=1);       /*等待发送*/
        for(i=16; i; i--)
            {
                      SBUF=*p;       /*发送数据*/
                      j=j + *p;      /*形成累加和*/
                      p++;
                      while(TI !=1);     /*等待发送*/
                 }
        SBUF=j;                 /*发送效验码*/
      j=0;
      while(TI !=1);
      while(RI !=1);       /*等待乙机回答*/
      if (SBUF==0)         /*若乙机正确接收，则返回，否则重发*/
           return;
      else
           goto sendDT;
}

main()
{
   com_init();
   send();
 }
```

乙机接收程序：**以下是 C 语言的程序[文件名：005B.c]**

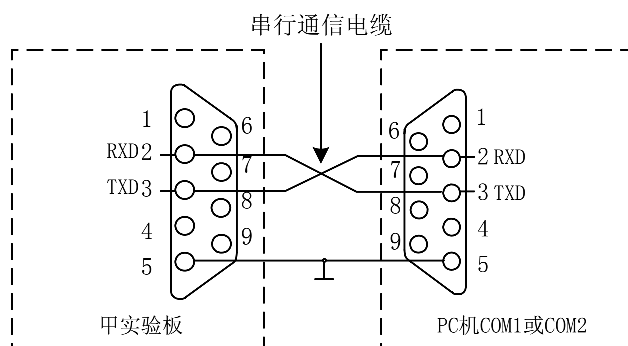
```
#include <reg51.h>
#define uchar unsigned char
uchar *p=0x20;              /*指针 p 指向内部 RAM 20h 单元*/
void com_init()             /*串口初始化程序*/
{
   TMOD=0x20;
   TH1=0xfa;
   TL1=0xfa;
   TR1=1;
   SCON=0x50;
}

reciv()
{
   uchar i;                 /*发送长度*/
   uchar j=0;         /*累加和清 0*/
recivDT:
   while( !RI);
   i=SBUF;           /*接收发送长度*/
   for(; i; i--)
     {
        while( !RI);
        *p=SBUF;
         p++;
         j=j+ *p;          /*计算累加效验和*/
     }
    while( !RI);            /*等待接收效验码*/
    if (j==SBUF)                /*接收正确则发送 0，否则发送 ff*/
       {
         SBUF=0x00;
         j=0;
         while( !RI);
         return;
        }
    else
       {
          SBUF=0xff;
          j=0;
          while( !RI);
          goto recivDT;
        }
}
```

```
main()
{
   com_init();
   reciv();
}
```

五、单片机和个人电脑（PC 机）的双机通信实验

用串行通信电缆线将实验平台的 9 针 D 型座和 PC 机的 COM1 和 COM2 连接起来（连线见下图）：



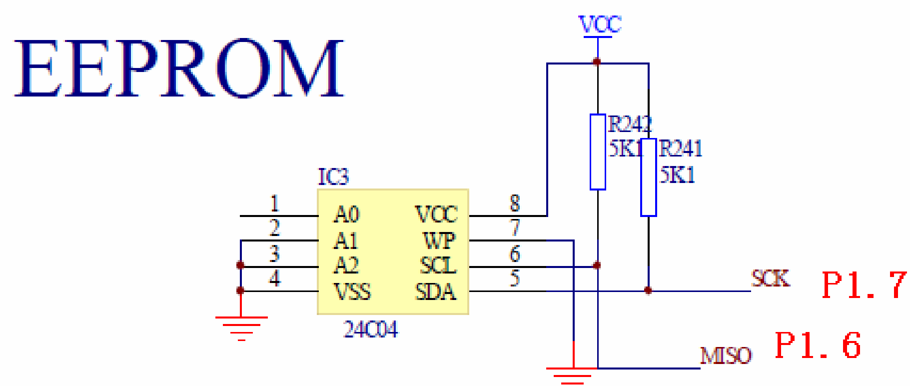
PC 上的程序可以用 Visual Basic 6.0 这样的开发工具调用 MSCOMM 控件进行串口通信，或者用 TC2.0 这样的 C 程序编译环境调用系统函数访问端口。如果您没有在 PC 上编写串口调试软件的能力，可以使用光盘中提供的“串口调试助手”与单片机方便的进行串口通信。

实验六　EEPROM 存储器实验[实验编号 006]

一、实验目的

1、掌握单片机扩展串行 EEPROM 方法。

2、掌握单片机对 I2C 串行接口的编程



二、实验内容和程序:

利用 24C04 断电以后存储的数据不消失的特点，可以做一个断电不间断计数器。

功能描述：单片机做一个 0－59 秒的自动计时器，用户可以随机关断系统电源，在通电以后计时器接着断电前的计数值继续计时。这个计数值是单片机把当前状态保存在 EEPROM 24C04 中的。

由于程序较为复杂，我们在光盘中提供 C 语言代码：**[文件名：006.c]**

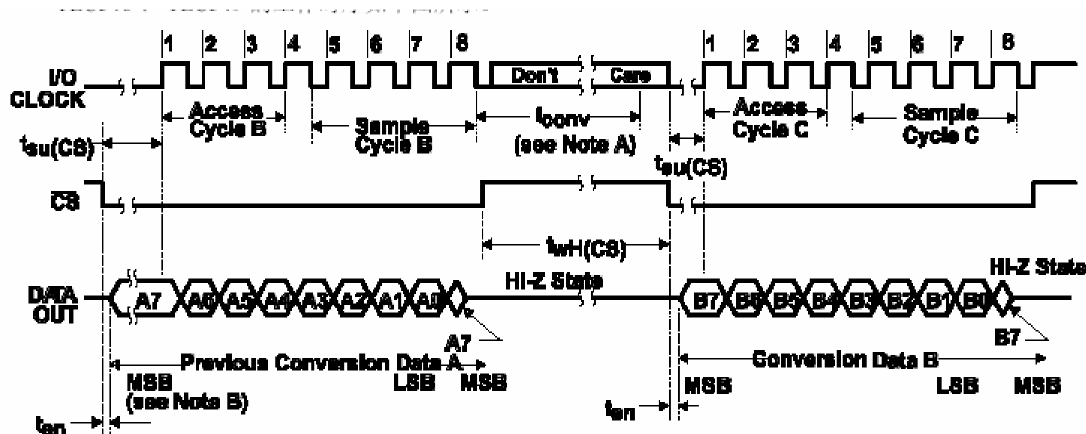
第七章 高级模块 A/D，D/A，液晶显示器说明

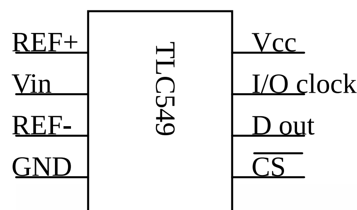
7．1　 串行 A/D 实验[实验编号 007]

实验板上的 AD 转换器采用了串行 8 位 A/D 转换器 TLC549

TLC548，TLC549 是美国德州仪器公司生产的 8 位串行 A／D 换器芯片，可与通用微处理器、控制器通过 I／O CLOCK、CS、DATA OUT 三线进行串行接口。具有 4MHz 片内系统时钟和软、硬件控制电路，转换时间最长 17us，TLC548 允许的最高转换速率为 45500 次／s，TLC549 为 40000 次／s。

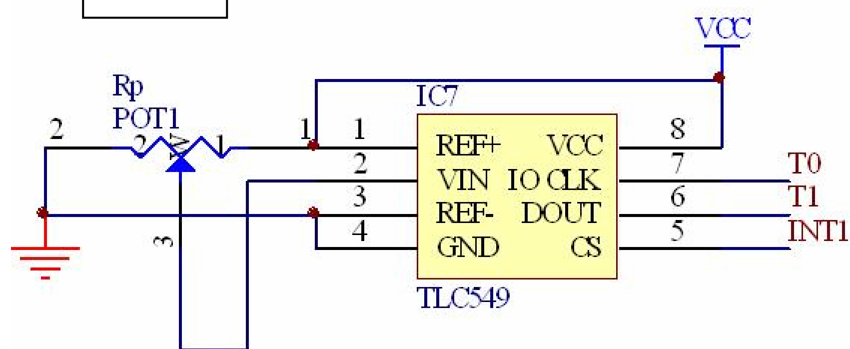
TLC549 时序和引脚见下图：





实验连线

实验连线见下图



实验内容和程序：

A/D tlc549 驱动程序，用 tlc549 采集可调电位器的电压，把电压转换为 0x00 到 0xFF 的 16 进制数字量，其中 0V 对应 0x00，5V 对应 0xFF

```
        org 0000h

ad:     setb p2.7       ;使蜂鸣器不发声
        mov a,#0
        setb p3.5       ;T1--DOUT
        setb p3.3       ;INT1--CS
        clr p3.4        ;T0--CLK
        clr p3.3
        acall read
        setb p3.3
        acall delay
        acall display
        sjmp ad

read:mov c,p3.5             ;读入转换数据一位
        rlc a           ;将进位左移给 A，即将转换的数据的一位读入
        mov r4, #07h

re0:    setb p3.4       ;置 I/O 时钟为高
        nop             ;等待转完
        nop
        clr p3.4        ;置 I/O 时钟为低
        nop
        nop
        mov c,p3.5          ;读入转换数据一位
        rlc a
        djnz r4,re0         ;8 位数据是否转完？
        setb p3.4
        nop
        nop
        clr p3.4
        nop
        nop

        ret

delay: mov r7,#05h
delay1: nop
        nop
        djnz r7,delay1
        ret
```

```
;***************
display:
        mov p1,#0
        mov r2,a

        mov a,r2
        anl a,#0f0h
        swap a
        mov dptr,#tab1
        mov r0,a          ;段码偏移
        mov r1,#01            ;位码

        mov a,r0
        movc a,@a+dptr            ;查到段码
        orl    a,#80h
        mov p2,a          ;送出段码，并使 p2.7=1,禁止蜂鸣器发声
        mov a,r1
        mov p1,a          ;送出位码
        mov r6,#40h
dl21:   mov r7,#5dh
dl11:   nop
        nop
        djnz r7,dl11
        djnz r6,dl21

        mov a,r2
        anl a,#0fh
        mov r0,a          ;找到下一个要显示的段码
        mov a,r1          ;位码
        rl a              ;位码左移
        mov r1,a

        mov a,r0
        movc a,@a+dptr            ;查到段码
        orl    a,#80h
        mov p2,a          ;送出段码，并使 p2.7=1,禁止蜂鸣器发声
        mov a,r1
        mov p1,a
                mov r6,#30h
dl22:   mov r7,#7dh
dl12:   nop
        ;nop
        djnz r7,dl12
        djnz r6,dl22
```

```
        reti
tab1:   db 3Fh,06h,5Bh,4Fh,66h,6Dh,7Ddh,07h,7Fh,6Fh,77h,7Ch,39h,5Eh,79h,71h
        end
```

与之功能相同的 C 语言代码在光盘中 **[文件名：009.c]**

7．2　　串行 D/A 实验[实验编号 008]

TLC5615是一个串行10位DAC芯片，只需要通过3 根串行总线就可以完成10位数据的串行传送，易于和工业标推的微处理器或微控制器(单片机)接口，适用于电池供电的测试仪表、移动电话，也适用于数字失调与增益调整以及工业控制场合。其引脚和时序如下图.

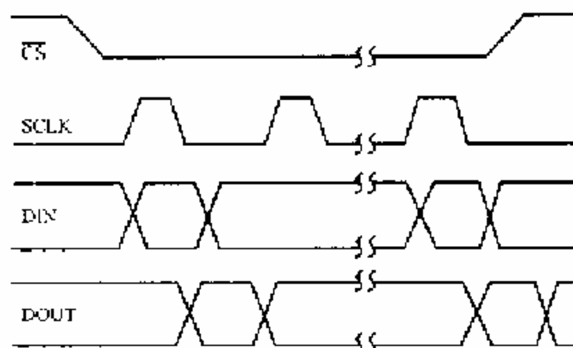
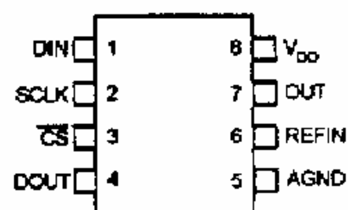
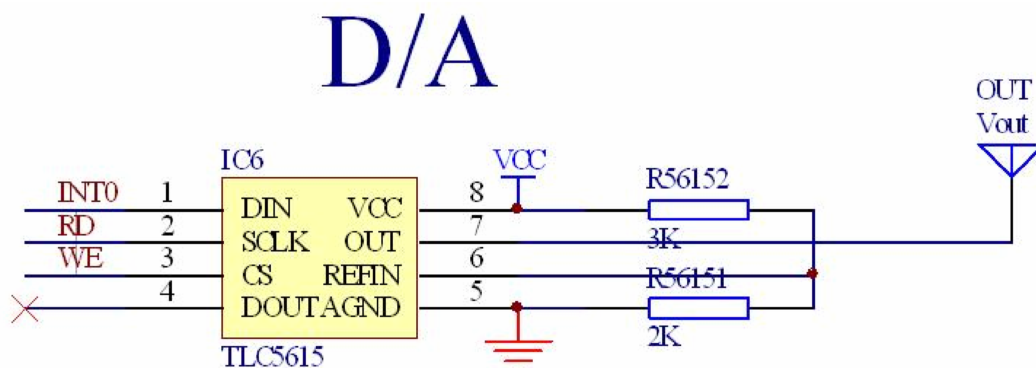


图 2　TLC5615 的时序图



5615 的引脚



实验连线见上图

实验内容和程序：

使 5615 产生锯齿波的驱动程序

以下是汇编语言的程序 **[文件名：008.asm]**

```
	org	0000h
	mov		r1,#0h
tlc5615:
	clr	p3.7
	clr	p3.2
	setb		p3.6 ;初始化，cs 为高电平
	nop
	nop
	inc	r1
	mov		r0,#0h
	clr	p3.6
	mov		a,r1
	mov		r3,#08h	;写 high 8 bit
	lcall		cuso
	mov		a,r0
	mov		r3,#04h
	lcall		cuso ;写 low 4 bit
	setb		p3.6
	sjmp		TLC5615
	ret
cuso:
	nop
loop:
	rlc	a
	mov		p3.2,c
	nop
	setb		p3.7
	nop
	clr	p3.7
	djnz		r3,loop
	ret
	end
```

与之功能相同的 C 语言代码在光盘中 **[文件名：008.c]**

7. 3 液晶显示器实验[实验编号 009]

我们的实验板配备了标准的液晶显示器接口，可以接驳各种标准接口的

◆主要技术参数:

显示容量:	16X2 个字符
芯片工作电压:	4.5～5.5V
工作电流:	2.0mA(5.0V)
模块最佳工作电压:	5.0V
字符尺寸:	2.95X4.35(WXH)mm

◆接口信号说明:

编号	符号	引脚说明	编号	符号	引脚说明
1	VSS	电源地	9	D2	Data I/O
2	VDD	电源正极	10	D3	Data I/O
3	VL	液晶显示偏压信号	11	D4	Data I/O
4	RS	数据/命令选择端（H/L）	12	D5	Data I/O
5	R/W	读/写选择端（H/L）	13	D6	Data I/O
6	E	使能信号	14	D7	Data I/O
7	D0	Data I/O	15	BLA	背光源正极
8	D1	Data I/O	16	BLK	背光源负极

1 基本操作时序：

1.1 读状态：输入：RS=L,RW=H,E=H 输出：D0～D7=状态字
1.2 写指令：输入：RS=L,RW=L,D0～D7=指令码,E=高脉冲 输出：无
1.3 读数据：输入：RS=H,RW=H,E=H 输出：D0～D7=数据
1.4 写数据：输入：RS=H,RW=L,D0～D7=数据,E=高脉冲 输出：无

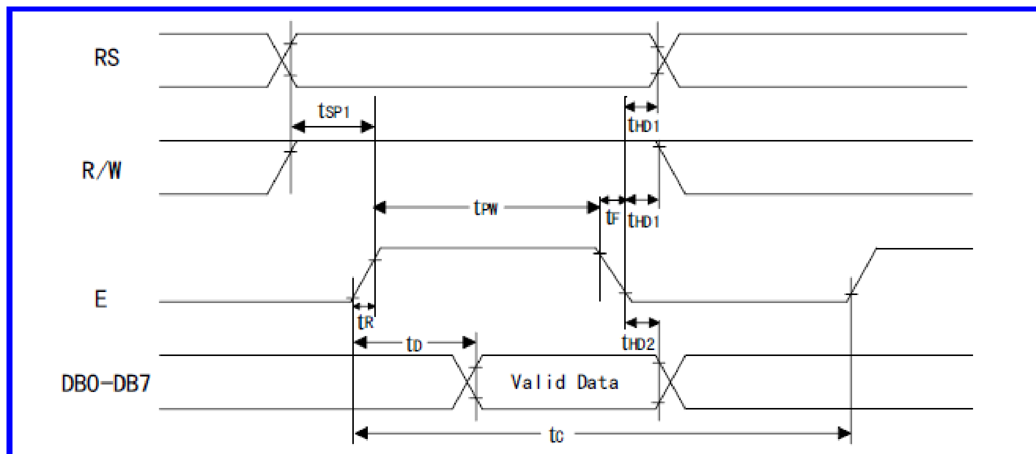
2 状态字说明

STA7 D7	STA6 D6	STA5 D5	STA4 D4	STA3 D3	STA2 D2	STA1 D1	STA0 D0

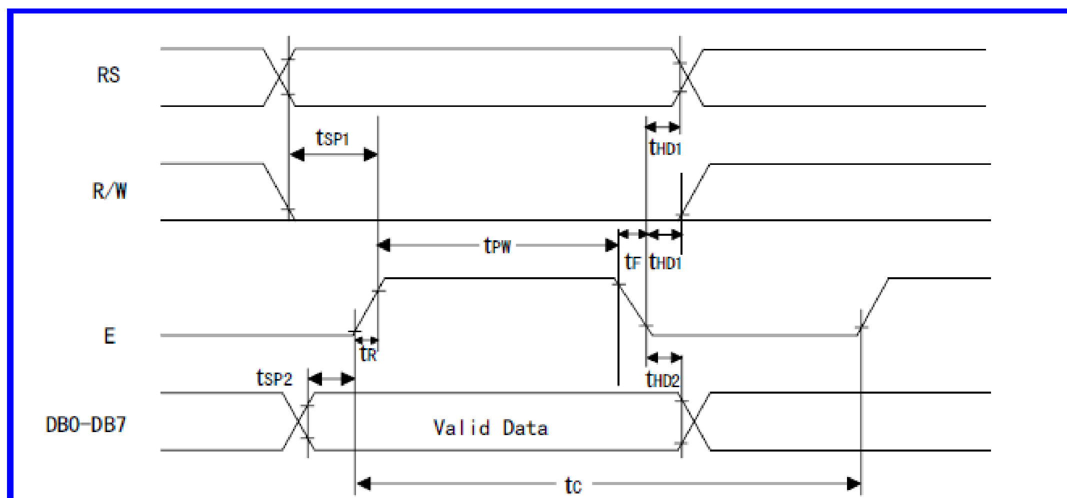
STA0-6	当前数据地址指针的数值	
STA7	读写操作使能	1: 禁止 0: 允许

注：对控制器每次进行读写操作之前，都必须进行**读写检测**，确保 STA7 为 0

1. 读操作时序



2. 写操作时序



实验内容和连线

内容描述：

在液晶显示器 1620 上显示两行字符：

详细说明和测试程序请在我们的光盘中**实验目录 009** 中

实验目录 009 中有以下几个文件：

lcd.c　　　液晶驱动程序 C 文件

lcd.h　　　定义液晶驱动管脚的头文件

main.c　　　测试程序主体 C 文件

编译这个程序时需要在 wave6000 环境或 Keil 环境中的工程菜单中加入这 3 个源文件才可以编译。

液晶显示器连线方法

LCD 引脚号	引脚功能	程序中对应的单片机引脚
1	GND	（接电源 GND）
2	VCC	（接电源 VCC）
3	VL	（接一个 500 欧姆的电阻到 GND）
4	RS - > P1.2	
5	RW - > P1.1	
6	E - > P1.0	
7	D0 -〉 P2.0	
8	D1 -〉 P2.1	
9	D2 -〉 P2.2	
10	D3 -〉 P2.3	
11	D4 -〉 P2.4	
12	D5 -〉 P2.5	
13	D6 -〉 P2.6	
14	D7 -〉 P2.7	
15	BLA	（接电源 VCC）
16	BLK	（接电源 GND）

第八章　SoftICE 在线仿真功能详细介绍

8.1 概述

8.1.1 介绍

SoftICE 的英文是 Software In Circuit Emulator 翻译成中文是：在线的软件仿真器。它是 SST 公司为方便 SST 用户使用 SST 单片机包括 SST89C5xSST89x564RD/554RC/516RD2/5XRD2 调试程序，所开发的开发工具。这个 SST 的仿真器功能可以代替大部分的专用仿真器的功能，使单片机工程师在开发单片机时省却了价格昂贵的仿真器和编程器使得 51 单片机的开发更加容易和普及。并且 SST 的单片机价格便宜出厂时就包含了在线下载和在线仿真的功能。本文是给购买了 AT89S52MCU 单片机仿真模块的用户一个详细使用说明
提醒：
单片 SST89XXX 制作的仿真器的特点是：
1，SST 公司没有提供源代码，公用监控程序不完善，有缺陷。
2，通信协议采用查询，不是采用中断，又没有异常处理措施，极易死机，麻烦多多（这也是为什么有些用一片 SST89C58（或其它替换品)制作的仿真器须在硬件上增加一复位键的原因）。

例如：

A，从开始调试→停止调试→开始调试——死机。
B，再次编译——死机。
C，从一工程到另一工程到编译——死机。

所以：我们不提供调试过程中不稳定情况的售后支持，用户可以经过摸索，避免出现调试过程中的死机情况。

AT89S52MCU 单片机增强板 仿真模块配备的是 SST89e564RD。

8.1.2 最新的软件/文件下载

最新的 SST 软件和文件请访问 SST 网页 http://www.sst.com 或 http://www.superflash.com

最新的 KEIL 软件和文件请访问 KEIL 网页 http://www.keil.com

8.1.3 SoftICE 的特点

SST 的 MCU SoftICE 通过 PC 的一个 COM 口与 KEIL uVision2 Debugger 通讯，它可以实时地调试目标程序，因此提供使用 SST 单片机的工程师简单有效和容易使用在板上调试程序。尽管小而紧凑，SoftICE 却提供高级仿真器的大部分功能。与 KEIL uVision2 Debugger 一起使用 SoftICE 提供以下特性：

▼ 源代码调试支持汇编语言和 C51 高级语言

▼ 单步执行 STEP 和 STEP OVER

▼ 断点调试做多到 10 个固定和 1 个临时断点

▼ 全速运行

▼ 显示修改变量

▼ 读/写数据存储器

▼ 读/写代码存储器

▼ 读/写 SFR 特殊功能寄存器

▼ 读/写 P0-P3 端口

▼ 下载 INTEL HEX 文件

▼ 对 8051 程序存储区的反汇编

▼ 在线汇编

▼ SST MCU 产品特有的 IAP 功能 In Application Programming

8.1.4 SoftICE 用到的 MCU 硬件资源

SST 的 SoftICE 用到的 MCU 硬件资源如下

(1) 一个 UART 串口 RXD TXD 用定时器 2 做为波特率发生器

(2) 8 个字节的堆栈空间

(3) SoftICE 监控程序分别占用下面 MCU 的 5Kbyte 程序空间

对 SST89C58 BLOCK1 的 1Kbyte 从 F000h 到 F3FFh

BLOCK0 的 4Kbyte 从 7000h 到 7FFFh

对 SST89C54 BLOCK1 的 1Kbyte 从 F000h 到 F3FFh

BLOCK0 的 4Kbyte 从 3000h 到 3FFFh

对 SST89x554RC BLOCK1 的 4KByte 从 0000h 到 0FFFh

BLOCK0 的 1Kbyte 从 7C00h 到 7FFFh

对 SST89x564RD BLOCK1 的 4Kbyte 从 0000h 到 0FFFh

BLOCK0 的 1Kbyte 从 FC00h 到 FFFFh

对 SST89E/V54RDX BLOCK1 的 1Kbyte 从 E000h 到 E3FFh

BLOCK0 的 4Kbyte 从 3C00h 到 3FFFh

对 SST89E/V58RDX BLOCK1 的 1Kbyte 从 E000h 到 E3FFh

BLOCK0 的 4Kbyte 从 7C00h 到 7FFFh

对 SST89E/V516RDX BLOCK1 的 4Kbyte 从 0000h 到 0FFFh

BLOCK0 的 1Kbyte 从 FC00h 到 FFFFh

其它部分的存储区域可以由用户的应用程序使用

8.2 进入调试状态

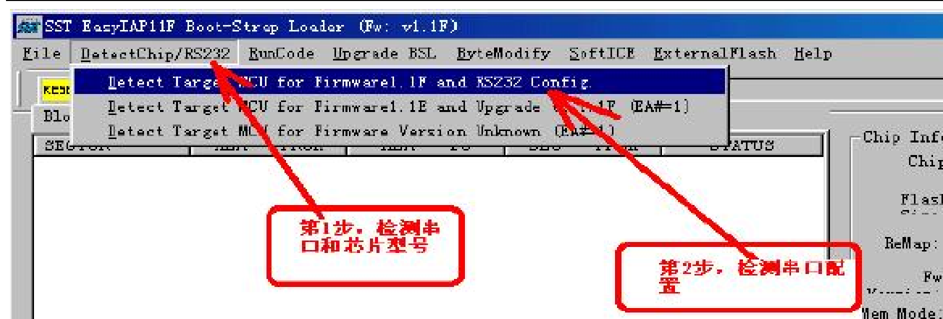
8.2.1 通过 SST 串口下载软件 BootLoader 下载 SOFTICE 监控代码

由于 SST 的 MCU 在出厂时已经将 BOOT LOADER 的下载监控程序写入到芯片中，因此无需编程器就可通过 SST BOOT-STRAP LOADER 软件工具将用户程序下载到 SST 的 MCU 中，从而运行用户程序。

SST BOOT-STRAP LOADER 软件工具还可将原来的 MCU 内部的下载监控程序转换为 SoftICE 的监控程序，从而实现 SOFTICE 的仿真功能。

执行 SSTEasyIAP11F.exe(在 **AT89S52MCU 光盘**的 SST 目录中)软件运行 SST Boot-Strap Loader ，在内部模式下检测到对应器件的型号后 SoftICE 固件通过按 SoftICE 菜单下 Download SoftICE 选项下载，便将 SoftICE 固件下载到 MCU ，在 BLOCK1 的 SST Boot-Strap Loader 会被 SoftICE 固件代替。详细操作步骤如下：

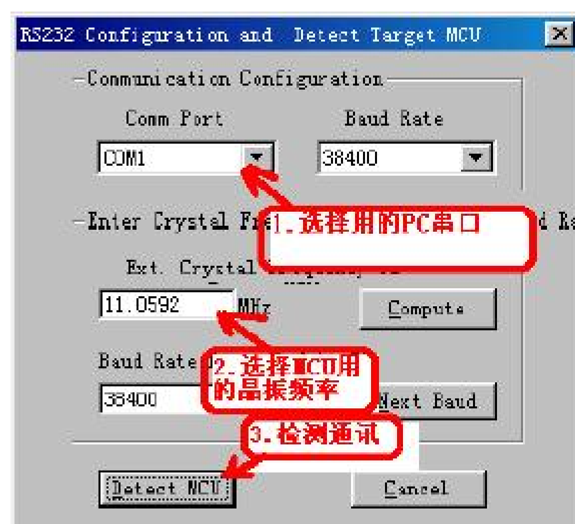
A 选择连接的串口
图 1



B 选择芯片型号和内部存储器模式

C 选择串口波特率 MCU 的晶振频率

图 3



D 按确定后再重新复位 MCU

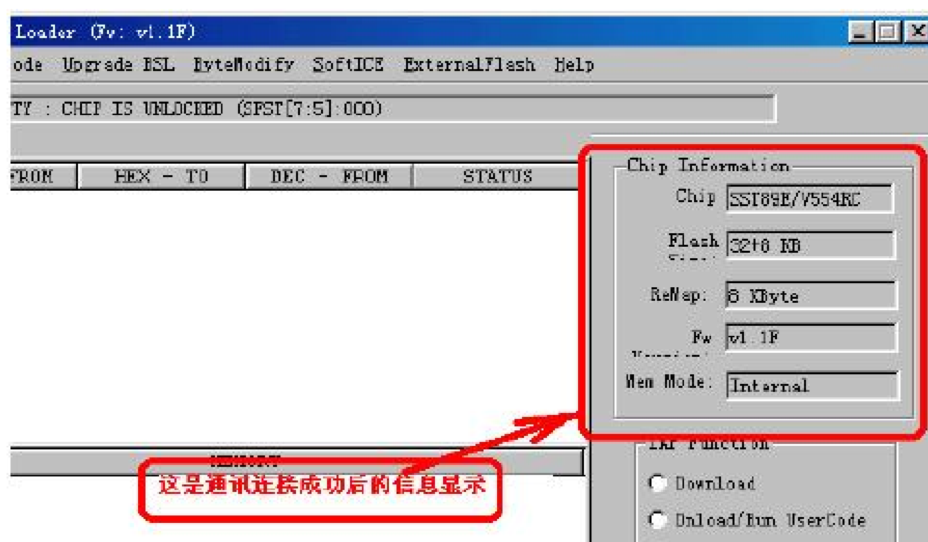
图 4



1.先按"确定"按钮,
2.然后再复位板上的MCU或重新上电.
3.这样才能通讯连接上.

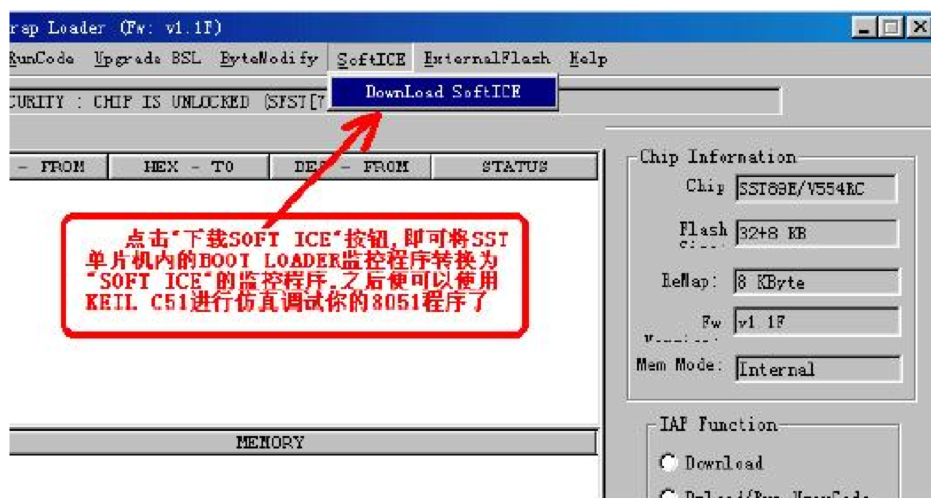
E 如果通讯成功在窗口的右上角可以显示出芯片的型号及 BOOT LOADER 的版本信息

图 5



F 点击 DOWNLOAD SOFTICE 的按钮即可将 MCU 内部的 BOOT LOADER 的监控程序更换为 SOFTICE 的监控程序

图 6



G 点击确定按钮确认转换为 SOFT ICE 的监控程序

图 7



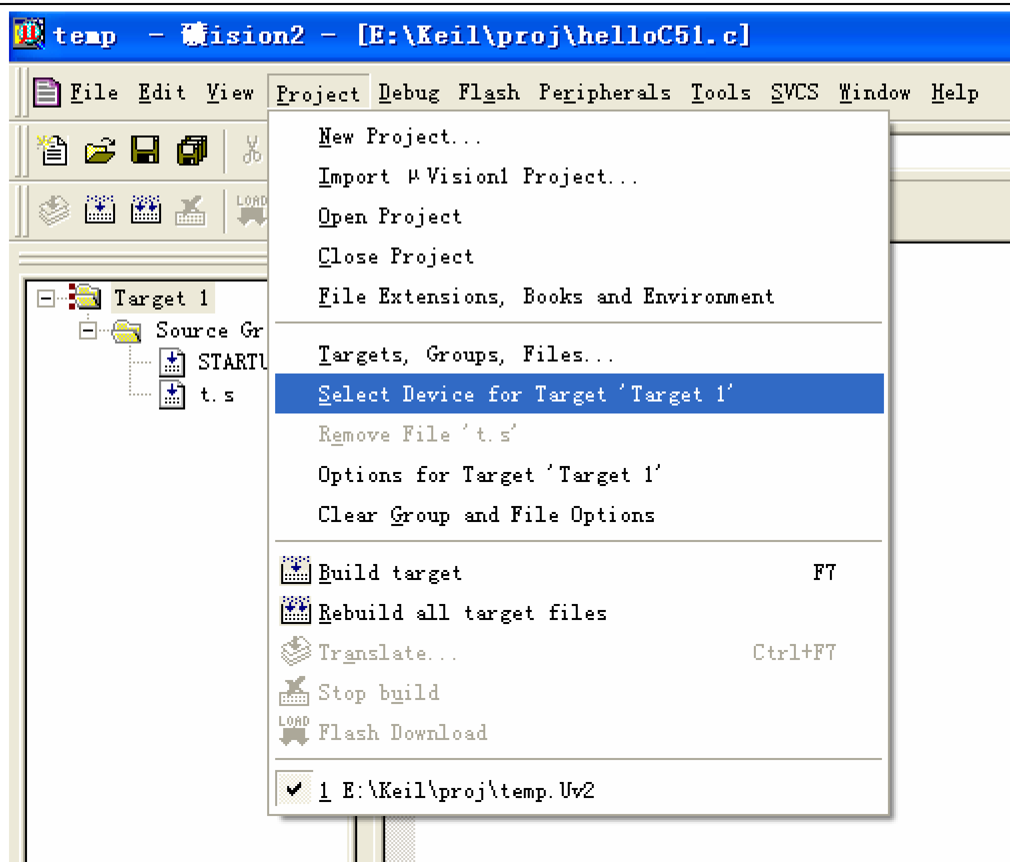
H 这是转换后的提示信息说明转换 SOFTICE 监控程序成功
图 8



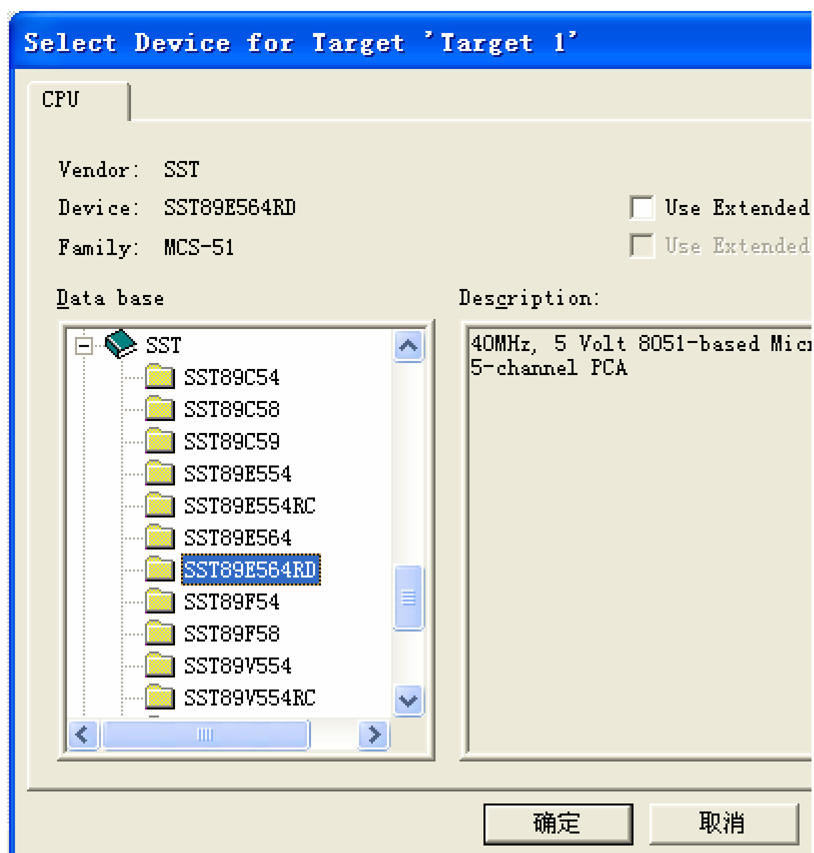
I 此时这片 SST89E554 芯片便具备了 SOFTICE 的在线仿真的功能

8.2.2 在 Keil 下，对 SST 芯片进行硬件调试的设置步骤

进入 Keil，进行如下设置；
在建好工程以后，选择



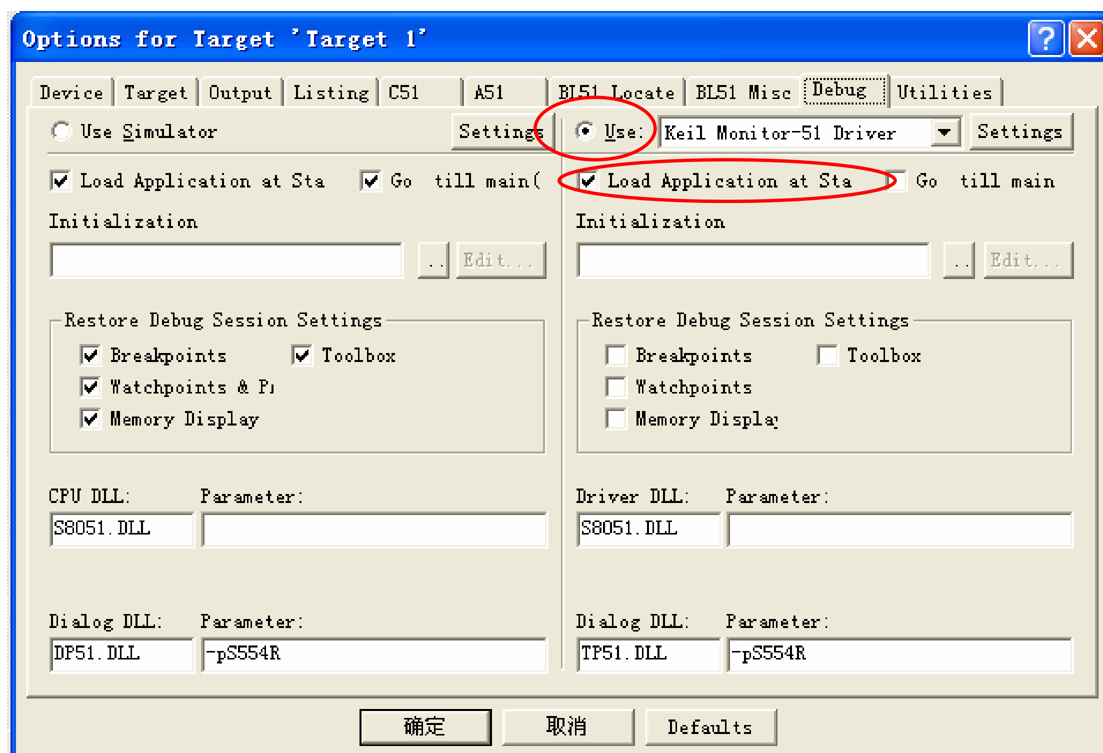
进入器件选择界面，选择你所使用的 SST 芯片型号：



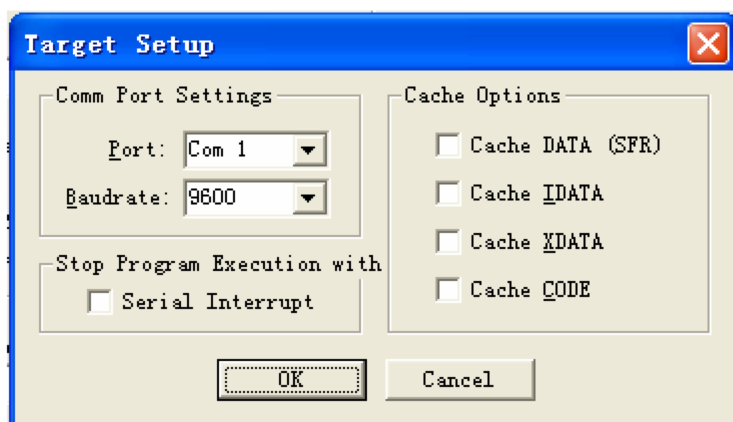
然后，开始设置调试选项，打开 project 菜单中的 Option for target “Target1” 选项，弹出如下菜单：

选择 debug 栏，选中右边的 Use：keil Monitor-51 Driver 项，同时下方的 Load

Application..也选中。



点击是 Settings，弹出菜单：



做如图所示选择，一般不选中 Cache 选项。

OK，确定以后，调试器的设置就完成了。

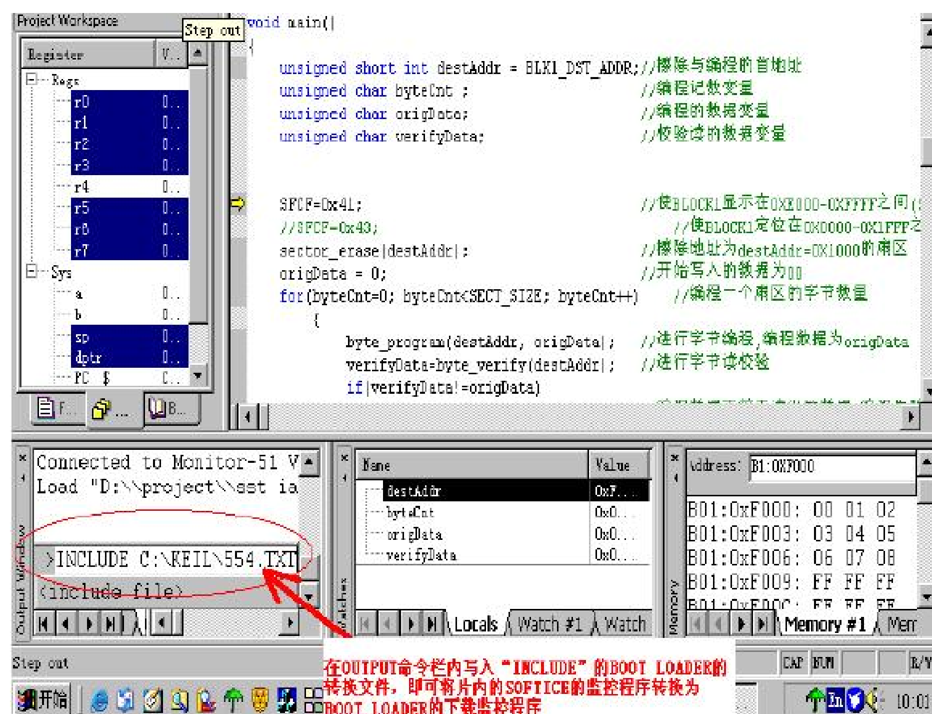
连好串口线，点击工具栏上的，开启调试器。

下面可以进入，SoftICE 的调试了。

8.3 由调试状态切换回下载状态

由 **SOFTICE** 转回到 **BOOTLOADER** 的步骤

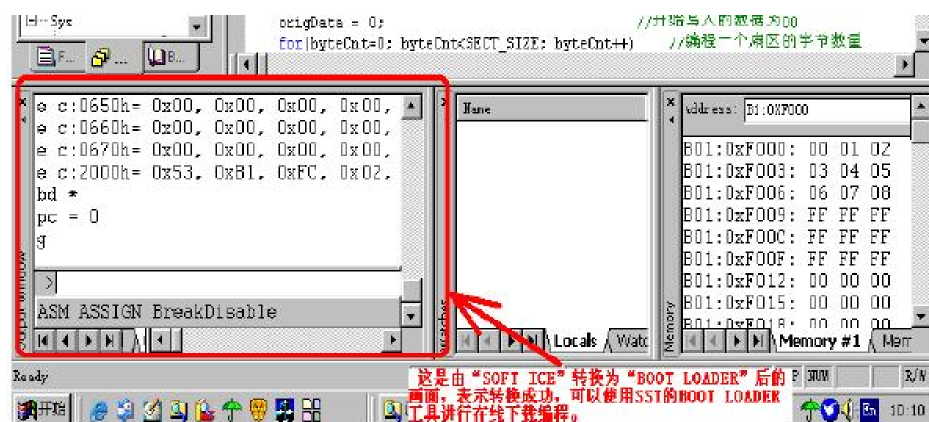
在 OUTPUT 窗口的命令行中写入 INCLUDE 具体芯片的转换文件名的命令回车
图 9



这些芯片的转换文件随 SST 每个芯片的 SOFTICE 工具包一起提供，可免费到 SST 的网站上下载。例如：下载的 SoftICE_58RD2.zip 工具包中就包含了一个 Convert_to_BSLx5xRD2.txt 文件。这个文件就是 SST89X58RD2 的内部监控程序由 SOFTICE 转换为 BOOT LOADER 的文件。

我们在光盘中提供了和我们的单片机板子配套的 SST 芯片的状态转换文件。在**光盘的 SST 目录**中，我们提供了一个 Convert_to_BSLx564.txt 文件，这个文件就是和 **AT89S52MCU 单片机仿真模块**配套的文件

执行完这个 INCLUDE Convert_to_BSLx5xRD2.txt 命令后，出现以下的画面表示转换为 BSL 监控程序成功。
图 10



小窍门：调试SST单片机特有的IAP（In Application Programming）功能

- 读/写SST特有的特殊功能寄存器SFR

要显示SST特有的特殊功能寄存器SFR，只需要在Watch Window窗口输入特殊功能寄存器的名称，Watch Window窗口在VIEW 菜单的点击Watch & Call Stack Window来激活，激活后就显示出SST特殊功能寄存器的当前值。

- SST特殊功能寄存器SFR可以在View菜单下Output Window中进行修改。

注：用户程序中要正确地定义SST的特殊功能寄存器。